**4-Year B. Tech in Computer Science & Engineering for in-house Course - Syllabus**

**1st Year Curriculum for B. Tech in Computer Science & Engineering for in-house Course**
*(Applicable from the academic session 2020-2021)*

## A. Definition of Credit:

| | |
|---|---|
| 1 Hr. Lecture (L) per week | 1 credit |
| 1 Hr. Tutorial (T) per week | 1 credit |
| 1 Hr. Practical (P) per week | 0.5 credits |

## B. Range of credits :

A range of credits from 150 to 160 for a student to be eligible to get B. Tech Degree in Engineering. A student will be eligible to get B. Tech Degree *with Honours*, if he/she completes an additional 20 credits. These could be acquired through Massive Open Online Courses (MOOCs).

## C. MOOCs for B. Tech Honours

The additional 20 credits (for obtaining B. Tech with Honours) are to be gained through MOOCs. The complete description of the MOOCs relevant for the first year course are given in *Annexure-I.* The courses for subsequent years of study will be posted subsequently.

## D. Guidelines regarding Mandatory Induction Program for the new students

All concerned are requested to follow the guidelines given in *Annexure-II* (Notice dt.06/12/2017) concerning Mandatory Induction Program. The colleges/ Institute may also refer to the AICTE Model Curriculum for Undergraduate Degree Courses in Engineering & Technology (January 2018) -Volume I (Page No.31-38), if necessary.

## E. Mandatory Additional Requirement for earning B. Tech Degree

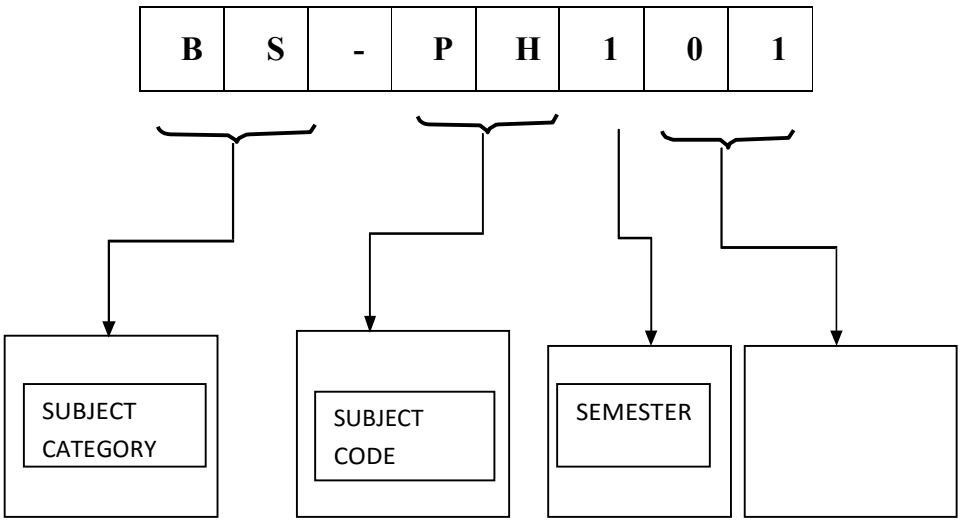
All concerned are requested to follow the guidelines in *Annexure-III* concerning Mandatory Additional Requirements.

## F. Codes for Departments

1. **Computer Science & Engineering ---code CSE**

## G. Subject Numbering Scheme:

**Specific code for the subject category**




| **List of Codes for Subject Category** | |
|---|---|
| Code | Category Name |
| BS | Basic Science Courses |
| ES | Engineering Science Courses |
| HSMC | Humanities and Social Sciences including Management courses |
| PCC | Professional core courses |
| PEC | Professional Elective courses |
| OEC | Open Elective courses |
| MC | Mandatory courses |
| PROJ | Project |

# First Year First Semester

## Mandatory Induction Program- 3 weeks duration

| Sl. No | Category | Subject Code | Subject Name | Total Number of contact hours | | | Credits |
|---|---|---|---|---|---|---|---|
| | | | | L | T | P | |
| **Theory** | | | | | | | |
| **1** | Basic Science course | BS-CH101 | Chemistry-I | 3 | 1 | 0 | 4 |
| **2** | Basic Science course | BS-M101 | Mathematics –I (Calculus & Linear Algebra) | 3 | 1 | 0 | 4 |
| **3** | Engineering Science Courses | ES-EE101 | Basic Electrical & Electronics Engineering | 4 | 0 | 0 | 4 |
| **3** | Engineering Science Courses | ES-ME101 | Engineering Graphics & Design | 1 | 0 | 0 | 1 |
| **Total Theory** | | | | **10** | **3** | **0** | **13** |
| **Practical** | | | | | | | |
| **1** | Basic Science course | BS-CH191 | Chemistry-I Lab | 0 | 0 | 4 | 2 |
| **2** | Engineering Science Courses | ES-EE191 | Basic Electrical & Electronics Engineering Lab | 0 | 0 | 2 | 1 |
| **3** | Engineering Science Courses | ES-ME191 | Engineering Graphics & Design | 0 | 0 | 4 | 2 |
| **Total Practical** | | | | **0** | **0** | **10** | **5** |
| **Total of First Semester** | | | | **10** | **3** | **10** | **18** |

## First Year Second Semester

| Sl. No | Category | Subject Code | Subject Name | Total Number of contact hours | | | Credits |
|---|---|---|---|---|---|---|---|
| | | | | L | T | P | |
| **Theory** | | | | | | | |
| **1** | Basic Science course | BS-PH201 | Physics-I | 3 | 1 | 0 | 4 |
| **2** | Basic Science course | BS-M201 | Mathematics –II (Probability and Statistics) | 3 | 1 | 0 | 4 |
| **3** | Engineering Science Courses | ES-CS201 | Programming for Problem Solving | 3 | 0 | 0 | 3 |
| **4** | Engineering Science Courses | ES-ME201 | Workshop/Manufacturing Practices | 1 | 0 | 0 | 1 |
| **5** | Humanities and Social Sciences including Management courses | HSMC 201 | English | 2 | 0 | 0 | 2 |
| **Total Theory** | | | | **12** | **2** | **0** | **14** |
| **Practical** | | | | | | | |
| **1** | Basic Science courses | BS-PH291 | Physics-I Lab | 0 | 0 | 4 | 2 |
| **2** | Engineering Science Courses | ES-CS291 | Programming for Problem Solving Lab | 0 | 0 | 4 | 2 |
| **3** | Engineering Science Courses | ES-ME291 | Workshop/Manufacturing Practices | | 0 | 4 | 2 |
| **4.** | Humanities and Social Sciences including Management courses | HSMC 291 | Language Lab | 0 | 0 | 2 | 1 |
| **Total Practical** | | | | **0** | **0** | **14** | **7** |
| **Total of Second Semester** | | | | **12** | **2** | **14** | **21** |

**Computer Science & Engineering**
**Semester-I (1st Year)**

| | |
|---|---|
| **Course Code** : BS-CH101 | **Category :** Basic Science Courses |
| **Course Title** : Chemistry-I | **Semester** : First |
| **L-T-P** : **3-1-0** | **Credit:4** |
| **Pre-Requisites:** | |

*Detailed contents*

**i) Atomic and molecular structure (10 lectures)**

Schrodinger equation. Particle in a box solutions and their applications for simple sample. Molecular orbitals of diatomic molecules (e.g. H2). Energy level diagrams of diatomic. Pi-molecular orbitals of butadiene and benzene and aromaticity. Crystal field theory and the energy level diagrams for transition metal ions and their magnetic properties. Band structure of solids and the role of doping on band structures.

**ii) Spectroscopic techniques and applications (8 lectures)**

Principles of spectroscopy and selection rules. Electronic spectroscopy. Fluorescence and its applications in medicine. Vibrational and rotational spectroscopy of diatomic molecules. Applications. Nuclear magnetic resonance and magnetic resonance imaging, surface characterisation techniques. Diffraction and scattering. **iii) Intermolecular *forces and potential energy surfaces (4 lectures)***

Ionic, dipolar and van Der Waals interactions. Equations of state of real gases and critical phenomena.

**iv) Use of free energy in chemical equilibria (8 lectures)**

First and second laws of thermodynamics and thermodynamic functions: energy, entropy and free energy. Estimations of entropy and free energies. Free energy and emf. Cell potentials, the Nernst equation and applications. Acid base, oxidation reduction and solubility equilibria. Water chemistry. Corrosion. Use of free energy considerations in metallurgy through Ellingham diagrams.

**v) Periodic properties (4 Lectures)**

Effective nuclear charge, penetration of orbitals, variations of s, p, d and f orbital energies of atoms in the periodic table, electronic configurations, atomic and ionic sizes, ionization energies, electron affinity and electronegativity, polarizability,

oxidation states, coordination numbers and geometries, hard soft acids and bases, molecular geometries

***vi)* Stereochemistry *(4 lectures)***

Representations of 3 dimensional structures, structural isomers and stereoisomers, configurations and symmetry and chirality, enantiomers, diastereomers, optical activity, absolute configurations and conformational analysis. Isomerism in transitional metal compounds.

***vii)* Organic *reactions and synthesis of a drug molecule (4 lectures)***

Introduction to reactions involving substitution, addition, elimination, oxidation, reduction, cyclization and ring openings. Synthesis of a commonly used drug molecule.

***Course Outcomes***

The concepts developed in this course will aid in quantification of several concepts in chemistry that have been introduced at the 10+2 levels in schools. Technology is being increasingly based on the electronic, atomic and molecular level modifications. Quantum theory is more than 100 years old and to understand phenomena at nanometer levels, one has to base the description of all chemical processes at molecular levels. The course will enable the student to:

- Analyse microscopic chemistry in terms of atomic and molecular orbitals and intermolecular forces.
- Rationalise bulk properties and processes using thermodynamic considerations.
- Distinguish the ranges of the electromagnetic spectrum used for exciting different molecular energy levels in various spectroscopic techniques
- Rationalise periodic properties such as ionization potential, electronegativity, oxidation states and electronegativity.
- List major chemical reactions that are used in the synthesis of molecules.

**Learning Resources:**

1. University chemistry, by B. H. Mahan
2. Chemistry: Principles and Applications, by M. J. Sienko and R. A. Plane
3. Fundamentals of Molecular Spectroscopy, by C. N. Banwell
4. Engineering Chemistry (NPTEL Web-book), by B. L. Tembe, Kamaluddin and M. S. Krishnan
5. Physical Chemistry, by P. W. Atkins
6. Spectroscopy of Organic Compounds, by P.S.Kalsi, New Age International Pvt Ltd Publishers

7. Physical Chemistry, P. C. Rakshit, Sarat Book House
8. Organic Chemistry: Structure and Function by K. P. C. Volhardt and N. E. Schore, 5th Edition http://bcs.whfreeman.com/vollhardtschore5e/default.asp
9. Engineering Chemistry, Satyaprakash, Khanna Book Publishing, Delhi

| | |
|---|---|
| **Course Code** : BS-M101 | **Category :** Basic Science Course |
| **Course Title** : Mathematics – I | **Semester** : First |
| **L-T-P** : **3-1-0** | **Credit: 4** |
| **Pre-Requisites: High School Mathematics** | |

| **Module No.** | **Description of Topic** | **Lectures Hours** |
|---|---|---|
| 1 | ***Calculus (Integration):*** Evolutes and involutes; Evaluation of definite and improper integrals; Beta and Gamma functions and their properties; Applications of definite integrals to evaluate surface areas and volumes of revolutions. | 8 |
| 2 | ***Calculus (Differentiation):*** Rolle's Theorem, Mean value theorems, Taylor's and Maclaurin's theorems with remainders; Indeterminate forms and L'Hospital's rule; Maxima and minima. | 6 |
| 3 | ***Matrices:*** Matrices, Vectors: addition and scalar multiplication, matrix multiplication; Linear systems of equations, linear Independence, rank of a matrix, determinants, Cramer's Rule, inverse of a matrix, Gauss elimination and Gauss-Jordan elimination. | 7 |

| | | |
|---|---|---|
| 4 | ***Vector Spaces:***<br>Vector Space, linear dependence of vectors, Basis, Dimension; Linear transformations (maps), Range and Kernel of a linear map, Rank and Nullity, Inverse of a linear transformation, Rank-Nullity theorem, composition of linear maps, Matrix associated with a linear map. | 9 |
| 5 | ***Vector Spaces (Continued):***<br>Eigenvalues,Eigenvectors, Symmetric, Skew-symmetric, and Orthogonal Matrices, Eigenbases.<br>Diagonalization; Inner product spaces, Gram-Schmidt orthogonalization. | 10 |

**Course Outcomes:**

The students will be able to:

- Apply the concept and techniques of differential and integral calculus to determine curvature and evaluation of different types of improper integrals.
- Understand the domain of applications of mean value theorems to engineering problems.
- Learn different types of matrices, concept of rank, methods of matrix inversion and their applications.
- Understand linear spaces, its basis and dimension with corresponding applications in the field of computer science.
- Learn and apply the concept of eigen values, eigen vectors, diagonalisation of matrices and orthogonalization in inner product spaces for understanding physical and engineering problems

**Learning Resources:**


1. Erwin Kreyszig, Advanced Engineering Mathematics, John Wiley & Sons.
2. Michael Greenberg, Advanced Engineering Mathematics, Pearson.
3. B.S. Grewal, Higher Engineering Mathematics, Khanna Publishers.
4. Kanti B. Dutta, Mathematical Methods of Science and Engineering, Cenage Learning.
5. Veerarajan T., Engineering Mathematics for first year, Tata McGraw-Hill, New Delhi.
6. S.K. Mapa, Higher Algebra: Abstract and Linear, Sarat Book House Pvt.Ltd.
7. Hoffman and Kunze: Linear algebra, PHI.
8. Reena Garg, Engineering Mathematics-I, Khanna Publishers.

| | |
|---|---|
| **Course Code** : ES-EE101 | **Category :** Engineering Science Courses |
| **Course Title** : Basic Electrical and Electronics Engineering | **Semester** : First |
| **L-T-P** : **4-0-0** | **Credit: 4** |
| **Pre-Requisites:**Basic concepts of electrical equipments and electronics Components | |

| **COURSE OBJECTIVES** |
|---|
| **1.** Impart a basic knowledge of electrical quantities such as current, voltage, power, energy and frequency to understand the impact of technology in a global and societal context. |
| **2.** Provide working knowledge for the analysis of basic DC and AC circuits used inelectrical and electronic devices. |
| **3.** To explain the working principle, construction, applications of DC machines, AC machines & measuring instruments. |
| **4.** Highlight the importance of transformers in transmission and distribution of electric power. |
| **5.** This course provides the student with the fundamental skills to understand the basic of semiconductor and components like diode, transistor, FET, MOSFET and operational amplifier. |
| **6.** It will build mathematical and numerical background for design of electronics circuit & component value. |
| **7.** Students equipped with the knowledge and training provided in the course will be able to participate in design , development and operation in the different area of electronics system. |

**Module 1:**

**DC circuits**: Mesh analysis, Superposition theorem, Thevenin's and Norton's theorems, Maximum Power Transfer theorem, delta star and star delta transformation.

**Electrostatics**: Introduction to Electrostatics, Gauss' theorem, Concept of capacitance. Different types of capacitors – parallel plate and cylindrical electrode arrangement. Stored energy in capacitors. Series-parallel combination of capacitors.

**Electromagnetism:** Review of fundamental laws of electromagnetism, Force on current carrying conductors, Magnetic circuits, permeance, reluctance, BH loop, hysteresis and eddy current losses, Inductance, Introduction to electromagnetic induction.

**Module 2:**

**AC circuits:** Sinusoidal and other periodic waveforms, average value, rms value, form factor, peak factor, representation of alternating quantities by phasors, Single phase series and parallel R, L and C circuits, reactance and impedance, resonance, active power, reactive power, apparent power and power factor, concept of power factor improvement**.**

**Three phase circuits:** Introduction to balanced three phase systems, Concept of phase sequence, relationship between line and phase voltages in star and delta connected systems, two wattmeter method for power measurement in balanced three phase circuits.

**Module 3:**

**Electrical Machines:** Principle of operation of transformers. Introduction to DC generators and motors. Principles of Three Phase Alternators, and Three Phase Induction Motors.

**Module 4:**

**Semiconductor fundamentals:** Band structure of solids, Fermi-dirac distribution, Semiconductor elemental & compound, Intrinsic and extrinsic semiconductor,

concept of effective mass and hole, generation and recombination of carriers, carrier diffusion.

**P-n junction:** Energy band diagram in equilibrium, under forward and reverse bias, I-V characteristics, breakdown mechanisms.

**Semiconductor Diodes:** Zener diode, LED, 7-Segment display, Photodiode, Solar cell.

**Module 5:**

**Diode Circuits:** Ideal model, Clipper, Clamper, Half-wave rectifier, Full-wave rectifier, Filter, Zener voltage regulator. Transistor: Structure and operation of BJT, JFET, MOSFET.

**Transistor Circuits:** CE, CB, CC configurations, Input – output characteristics, biasing, loadline, Q-point analysis, Analysis of an amplifier using simplified h-parameter model. Direct coupled, RC coupled and transformer coupled amplifiers. Feedback Circuits and Oscillators: General theory of feedback, negative feedback and its advantages, requirement for oscillation, oscillators. Transistorized voltage regulator.

**Module 6:**

**IC and Op-amp Circuits:** Monolithic ICs, Analog/Digital/Hybrid ICs – basics, Ideal op-amp, Inverting amplifier, Non-inverting amplifier, Buffer amplifier, Summing amplifier, Difference amplifier, Differentiator; Integrator, Op-amp as a comparator, Square wave generator, Triangular wave generator.

**Logic Circuits:** Number systems, Boolean algebra, Basic gates, Simple circuits using gates, Transistor as a switch, CMOS inverter; Block diagram level descriptions – Multiplexer, Encoder, Decoder, Flip-flop, Register, Counter. Basic Electronic Measurements: Multimeter and CRO.

| LECTURE WITH BREAKUP | NO. OF LECTRES |
|---|---|
| **Module 1:**<br><br>**DC circuits**: Mesh analysis, Superposition theorem, Thevenin's and Norton's theorems, Maximum Power Transfer theorem, delta star and star delta transformation.<br><br>**Electrostatics**: Introduction to Electrostatics, Gauss' theorem, Concept of capacitance. Different types of capacitors – parallel plate and cylindrical electrode arrangement. Stored energy in capacitors. Series-parallel combination of capacitors.<br><br>**Electromagnetism:** Review of fundamental laws of electromagnetism, Force on current carrying conductors, Magnetic circuits, permeance, reluctance, BH loop, Hysteresis and eddy current losses, Inductance, Introduction to electromagnetic induction. | **[6]** |
| **Module 2:**<br><br>**AC circuits:** Sinusoidal and other periodic waveforms, average value, rms value, form factor, peak factor, representation of alternating quantities by phasors, Single phase series and parallel R, L and C circuits, reactance and impedance, resonance, active power, reactive power, apparent power and power factor, concept of power factor improvement**.**<br><br>**Three phase circuits:** Introduction to balanced three phase systems, Concept of phase sequence, relationship between line and phase voltages in star and delta connected systems, two wattmeter method for power measurement in balanced three phase circuits.<br><br>. | **[8]** |

| | |
|---|---|
| **Module 3:**<br><br>**Electrical Machines:** Principle of operation of transformers. Introduction to DC generators and motors. Principles of Three Phase Alternators, and Three Phase Induction Motors | **[10]** |
| **Module 4:**<br><br>**Semiconductor fundamentals:** Band structure of solids, Fermi-dirac distribution, Semiconductor elemental & compound, Intrinsic and mextrinsic semiconductor, concept of effective mass and hole, generation and recombination of carriers, carrier diffusion.<br><br>**P-n junction:** Energy band diagram in equilibrium, under forward and reverse bias, I-V characteristics, breakdown mechanisms.<br><br>**Semiconductor Diodes:** Zener diode, LED, 7-Segment display, Photodiode, Solar cell. | **[5]** |
| **Module 5:**<br><br>**Diode Circuits:** Ideal model, Clipper, Clamper, Half-wave rectifier, Full-wave rectifier, Filter, Zener voltage regulator. Transistor: Structure and operation of BJT, JFET, MOSFET.<br><br>**Transistor Circuits:** CE, CB, CC configurations, Input – output characteristics, biasing, loadline, Q-point analysis, Analysis of an amplifier using simplified h-parameter model. Direct coupled, RC coupled and transformer coupled amplifiers. Feedback Circuits and Oscillators: General theory of feedback, negative feedback and its advantages, requirement for oscillation, oscillators. Transistorised voltage regulator. | **[10]** |
| **Module 6:**<br><br>**IC and Op-amp Circuits:** Monolithic ICs, Analog/Digital/Hybrid ICs – basics, Ideal op-amp, Inverting amplifier, Non-inverting amplifier, Buffer amplifier, Summing amplifier, Difference amplifier, Differentiator; Integrator, Op-amp as a comparator, Square wave | **[9]** |

| | |
|---|---|
| generator, Triangular wave generator.<br><br>**Logic Circuits:** Number systems, Boolean algebra, Basic gates, Simple circuits using gates, Transistor as a switch, CMOS inverter; Block diagram level descriptions – Multiplexer, Encoder, Decoder, Flip-flop, Register, Counter. Basic Electronic Measurements: Multimeter and CRO. | |

| **COURSE OUTCOMES** |
|---|
| **After completion of course, students would be able to:** |
| CO1: To study basics of semiconductor &devices and their applications in different areas. |
| CO2: To study different biasing techniques to operate transistor, FET , MOSFET and operational amplifier in different modes. |
| CO3: Analyze output indifferent operating modes of different Semiconductor devices. |
| CO4: Identify the type of electrical machine used for that particular application. |
| CO5: Realize the requirement of transformers in transmission and distribution of electric power and other applications |
| CO6: Formulate and solve complex AC, Dc circuits |

**Textbooks/References:**

**Suggested books:**


1. Advanced Electrical Technology – H. Cotton
2. Electrical Technology – Hughes
3. Alternating Current Circuits – Kerchner and Corcoran
4. Fundamentals of Electrical Engineering – Ashfaq Husain
5. Solid State Electronic Devices by Ben G. Streetman and Sanjay K. Banerjee, Pearson Prentice Hall, 7th Edition, 2014.

**Suggested reference books:**


1. Electronic Circuits: Discrete and Integrated by D. L. Schilling and C. Belove, McGraw-Hill, 1989 Electronics
2. Fundamentals and Applications by D. Chattopadhyay and P. C. Rakshit, New Age International, 10th Edition, 2010
3. Digital Principles and Applications by A. P. Malvino and D. P. Leach, Tata McGraw-Hill, 7th edition, 2006
4. Electroni c Principles by Albert Malvino and David Bates, Tata McGraw-Hill, 7th edition, 2017


| | |
|---|---|
| **Course Code** : BS-CH191 | **Category :** Basic Science Courses |
| **Course Title** : Chemistry-I Lab | **Semester** : First |
| **L-T-P** : **0-0-4** | **Credit:2** |
| **Pre-Requisites:** | |

**Choose 10 experiments from the following:**

1. Conductometric titration for determination of the strength of a given HCl solution by titration against a standard NaOH solution.
2. pH- metric titration for determination of strength of a given HCl solution against a standard NaOH solution.
3. Determination of dissolved oxygen present in a given water sample.
4. To determine chloride ion in a given water sample by Argentometric method (using chromate indicator solution)
5. Determination of surface tension and viscosity
6. Thin layer chromatography
7. Ion exchange column for removal of hardness of water
8. Determination of the rate constant of a reaction
9. Determination of cell constant and conductance of solutions
10. Potentiometry - determination of redox potentials and emfs
11. Saponification/acid value of an oil

12. Chemical analysis of a salt
13. Determination of the partition coefficient of a substance between two immiscible liquids
14. Adsorption of acetic acid by charcoal
15. Use of the capillary viscosimeters to the demonstrate of the isoelectric point as the pH of minimum viscosity for gelatin sols and/or coagulation of the white part of egg .

| | |
|---|---|
| **Course Code** | **ES-EE191** |
| **Course Name** | **BASIC ELECTRICAL AND ELECTRONICS LAB** |
| **Credits** | **1 (0L:0T:2P)** |
| **Pre-Requisites** | **Basic concept of electrical and electronics equipments** |

**Total Number of Lectures: 24**

| **COURSE OBJECTIVES** |
|---|
| **1. To familiarize with the electrical component like inductors, capacitor, resistors, etc.** |
| **2. Understanding of various theorems into practical applications.** |
| **2. Better understanding of electrical components like diode, Operational amplifier and practical understanding of various characteristics of it.** |

**LIST OF EXPERIMENTS:**

1. Familiarization with Resistor, capacitor and Inductor
2. Study of VI characteristics of a Diode
3. Study of DC network theorem
4. Half Wave Rectifier and Full Wave Rectifier
5. Study of Series and parallel RLC circuit
6. Zener Diode-Voltage regulator
7. Study of DC Motor
8. Studies on BJT Common Base, Common Collector, Common

Emitter characteristics

9. Study of Three phase induction motor
10. Study of basic properties of operational amplifier, inverting and non-inverting amplifiers.
11. Study of differentiator and integrator using operational amplifier
12. Study of Transformer

| | |
|---|---|
| **Course Code** : ES-ME191 | **Category :** Engineering Science Courses |
| **Course Title** : Engineering Graphics & Design | **Semester** : First |
| **L-T-P** : **1-0-4** | **Credit: 3** |
| **Pre-Requisites:** | |

| **Sl. No.** | **Content** | **Lecture (L)** | **Practical (P)** |
|---|---|---|---|
| 1 | **INTRODUCTION TO ENGINEERING DRAWING**<br><br>Principles of Engineering Graphics and their significance, usage of Drawing instruments, lettering, Different types of lines and their use; Drawing standards and codes. | 1 | 4 |
| 2 | **LETTERING, DIMENSIONING, SCALES**<br><br>Plain scale, Diagonal scale and Vernier Scales. | 1 | 4 |
| 3 | **GEOMETRICAL CONSTRUCTION AND CURVES**<br><br>Construction of polygons, Conic sections including the Rectangular Hyperbola (General method only); Cycloid, Epicycloid, Hypocycloid, Involute, Archemedian Spiral. | 1 | 4 |
| 4 | **PROJECTION OF POINTS, LINES, SURFACES**<br><br>Principles of Orthographic Projections-Conventions - 1st and 3rd angle projection, Projections of Points and lines inclined to both planes; Projections of planes (Rectangle, | 1 | 4 |

| | | | |
|---|---|---|---|
| | pentagon, Hexagon etc.) inclined Planes<br>- Auxiliary Planes. | | |
| 5 | **PROJECTION OF REGULAR SOLIDS**<br><br>Regular solids inclined to both the Planes- Auxiliary Views; Draw simple annotation, dimensioning and scale (Cube, Pyramid, Prism, Cylinder, Cone). | 1 | 4 |
| 6 | **COMBINATION OF REGULAR SOLIDS, FLOOR PLANS**<br><br>Regular solids in mutual contact with each other like Spheres in contact with cones standing on their base. Floor plans that include: windows, doors, and fixtures such as WC, bath, sink, shower, etc. | 1 | 4 |
| 7 | **ISOMETRIC PROJECTIONS**<br><br>Principles of Isometric projection – Isometric Scale, Isometric Views, Conventions; Isometric Views of lines, Planes, Simple and compound Solids; Conversion of Isometric Views to Orthographic Views and Vice-versa, Conventions; | 1 | 4 |
| 8 | **SECTIONS AND SECTIONAL VIEWS OF RIGHT ANGULAR SOLIDS**<br>Prism, Cylinder, Pyramid, Cone – Auxiliary Views; Development of<br><br>surfaces of Right Regular Solids - Prism, Pyramid, Cylinder and Cone; Draw the sectional orthographic views of geometrical solids, objects from industry and dwellings (foundation to slab only) | 1 | 4 |
| | **OVERVIEW OF COMPUTER GRAPHICS, CUSTOMISATION&**<br><br>**CAD DRAWING** | | |

| | | | |
|---|---|---|---|
| 9 | listing the computer technologies that impact on graphical communication, Demonstrating knowledge of the theory of CAD software [such as: The Menu System, Toolbars (Standard, Object Properties, Draw, Modify and Dimension), Drawing Area (Background, Crosshairs, Coordinate System), Dialog boxes and windows, Shortcut menus (Button Bars), The Command Line (where applicable), The Status Bar, Different methods of zoom as used in CAD, Select and erase objects.; Isometric Views of lines, Planes, Simple and compound Solids]; Set up of the drawing page and the printer, including scale settings, S etting up of units and drawing limits; ISO and ANSI standards for coordinate dimensioning and tolerancing; Orthographic constraints, Snap to objects manually and automatically; Producing drawings by using various coordinate input entry methods to draw straight lines, Applying various ways of drawing circles; | 1 | 4 |
| | **ANNOTATIONS, LAYERING & OTHER FUNCTIONS**<br><br>applying dimensions to objects, applying annotations to drawings;<br><br>Setting up and use of Layers, layers to create drawings, Create, edit and use customized layers; Changing line lengths through modifying existing lines (extend/lengthen); Printing documents to paper using the print command; orthographic projection techniques; Drawing sectional views of composite right regular geometric solids and project the true shape of the sectioned surface; Drawing annotation, Computer- aided design (CAD) software modeling of parts and assemblies. Parametric and non-parametric solid, surface, and wireframe models. Part editing and two-dimensional documentation of models. Planar projection theory, including sketching of perspective, isometric, multiview, auxiliary, and section views. Spatial | | |

| | | | |
|---|---|---|---|
| | visualization exercises. Dimensioning guidelines, tolerancing techniques; dimensioning and scale mult i views of dwelling; | | |
| 11 | **DEMONSTRATION OF A SIMPLE TEAM DESIGN PROJECT** Geometry and topology of engineered components: creation of engineering models and their presentation in standard 2D blueprint form and as 3D wire-frame and shaded solids; meshed topologies for engineering analysis and tool-path generation for component manufacture; geometric dimensioning and tolerancing; Use of solid- modeling software for creating associative models at the component and assembly levels; floor plans that include: windows, doors, and fixtures such as WC, bath, sink, shower, etc. Applying colour coding according to building drawing practice; Drawing sectional elevation showing foundation to ceiling; Introduction to Building Information Modelling (BIM). | 2 | 8 |

**Course Outcomes**

The student will learn:

- Introduction to engineering design and its place in society
- Exposure to the visual aspects of engineering design
- Exposure to engineering graphics standards
- Exposure to solid modelling

**General Instructions**

1. In every topic some problems are to be done in the class and some are to be given to students as home assignment.
2. The problems for class work are to be prepared on drawing sheet of A1 size in the class/ using
AutoCAD software.
3. The problems for home assignments are to be prepared on drawing copy/ using AutoCAD software.
4. Print out of every assignment is to be taken for CAD Drawings on Drawing sheets (A4 Sheets).
5. A title block must be prepared in each sheet/ assignment.

Following is the list of drawing instruments that required for making engineering drawings on paper with perfection

1. Drawing Board
2. Mini drafter/ Set-squares (45°–45° & 60°–90°), T-square
3. Protractor (180°, 360°)
4. Scales (Plain, Diagonal)
5. Compass (Small and Large)
6. Divider (Small and Large)
7. French Curves
8. Drawing paper (A1 Size)
9. Drawing pencil (H, HB, B)
10. Sharpener
11. Eraser
12. Drawing pins & clips
13. Duster or handkerchief etc.

**Learning Resources:**

1. Bhatt N.D., Panchal V.M. & Ingle P.R., (2014), Engineering Drawing, Charotar Publishing House
2. Agrawal B. & Agrawal C. M. (2012), Engineering Graphics, TMH Publication
3. Shah, M.B. & Rana B.C. (2008), Engineering Drawing and Computer Graphics, Pearson Education
4. Narayana, K.L. & P Kannaiah (2008), Text book on Engineering Drawing, Scitech Publishers
5. Corresponding set of CAD Software Theory and User Manuals
6. Pradeep Jain, Ankita Maheswari, A.P. Gautam, Engineering Graphics & Design, Khanna Publishing House

**Computer Science & Engineering**
**Semester –II ( 1st Year)**

| | |
|---|---|
| **Course Code** :BS-PH201 | **Category :** Basic Science Courses |
| **Course Title** : Physics-I | **Semester** : Second |
| **L-T-P** : **3-1-0** | **Credit:4** |
| **Pre-Requisites:** | |

**Course objectives:**

Basic concepts of mechanics, optics and its applications, electricity, magnetism and qualitative understanding of concepts of quantum physics and statistical mechanics.

**1. Mechanics** ( 7L)

Problems including constraints & friction. Basic ideas of vector calculus and partial differential equations. Potential energy function F = -grad V, equipotential surfaces and meaning of gradient. Conservative and non-conservative forces. Conservation laws of energy & momentum. Non-inertial frames of reference. Harmonic oscillator; Damped harmonic motion forced oscillations and resonance. Motion of a rigid body in a plane and in 3D. Angular velocity vector. Moment of inertia.

**2. Optics** (5L)

□ Distinction between interference and diffraction, Fraunhofer and Fresnel diffraction, Fraunhofer diffraction at single slit, double slit, and multiple slits ( only the expressions for max;min, & intensity and qualitative discussion of fringes); diffraction grating(resolution formulac only), characteristics of diffration grating and its applications.
□ Polarisation : Introduction, polarisation by reflection, polarisation by double reflection, scattering of light, circular and elliptical polarisation, optical activity.
□ Lasers : Principles and working of laser : population inversion, pumping, various modes, threshold population inversion with examples .

**3. Electromagnetism and Dielectric Magnetic Properties of Materials** (8L)

□ Maxwell's equations. Polarisation, permeability and dielectric constant, polar and non-polar dielecrrics, internal fields in a solid, Clausius- Mossotti equation(expression only), applications of dielectrics.

□ Magnetisation , permeability and susceptibility, classificationof magnetic materials, ferromagnetism, magnetic domains and hysteresis, applications

**4. Quantum Mechanics** (16L)

□ Introduction to quantum physics, black body radiation, explanation using the photon concept, Compton effect, de Broglie hypothesis, wave-particle duality, verification of matter waves, uncertainty principle, Schrodinger wave equation, particle in box, quantum harmonic oscillator, hydrogen atom.

**5. Statistical Mechanics**(8L)

□ Macrostate, Microstate, Density of states, Qualitative treatment of Maxwell Boltzmann, Fermi-Dirac and Bose-Einstein statistics.

**Course outcomes:**

Students will be familiar with

□ Basic concepts of mechanics

□ Bragg's Law and introduction to the principles of lasers, types of lasers and applications.

□ Various terms related to properties of materials such as, permeability, polarization,etc.

□ Some of the basic laws related to quantum mechanics as well as magnetic and dielectric properties of materials.

□ Simple quantum mechanics calculations.

**Learning Resources:**

1. Introduction to Electrodynamics, David J. Griffiths, Pearson Education India Learning Private Limited
2. Principles of Physics, 10ed, David Halliday, Robert Resnick Jearl Walker , Wiley
3. Electricity, Magnetism, and Light, Wayne M. Saslow, Academic Press

4. Engineering Mechanics (In SI Units) (SIE), S. Timoshenko, D.H. Young, J.V. Rao, Sukumar Pati , McGraw Hill Education
5. Classical mechanics, Narayan Rana, Pramod Joag, McGraw Hill Education
6. Introduction to Classical Mechanics, R Takwale, P Puranik, McGraw Hill Education
7. Engineering Mechanics, M.K. Harbola , Cengage India
8. An Introduction to Mechanics (SIE), David Kleppner, Robert Kolenkow, McGraw Hill Education
9. Principles of mechanics, John L. Synge and Byron A. Griffith, New York, McGraw-Hill
10. Mechanics (Dover Books on Physics) , J. P. Den Hartog , Dover Publications Inc.
11. Engineering Mechanics: Dynamics, L.G. Kraige J.L. Meriam, Wiley
12. Quantum Physics of Atoms, Molecules, Solids, Nuclei and Particles, Robert Eisberg, Robert Resnick, Wiley
13. Introduction to Quantum Mechanics, J. Griffiths David , Pearson Education
14. Modern Quantum Mechanics, J. J. Sakurai, Cambridge University Press
15. Optics , Hecht, Pearson Education
16. Optics, Ghatak, McGraw Hill Education India Private Limited
17. Fundamentals of Statistical and Thermal Physics, Reif, Sarat Book Distributors
18. Statistical Mechanics , Pathria , Elsevier
19. Statistical Physics, L.D.Landau , E.M. Lifshitz, Butterworth-Heinemann

| | |
|---|---|
| **Course Code** : BS-M201 | **Category :** Basic Science Course |
| **Course Title** : Mathematics – II | **Semester** : Second |
| **L-T-P** : **3-1-0** | **Credit: 4** |
| **Pre-Requisites:** High School Mathematics and BS-M101 | |

| Module No. | Description of Topic | Lectures Hours |
|---|---|---|
| 1 | **Basic Probability:** Probability spaces, conditional probability, independence;<br><br>Discrete random variables, Independent random variables, the Multinomial distribution, Poisson approximation to the Binomial distribution, infinite sequences of Bernoulli trials, sums of independent random variables; Expectation of Discrete Random Variables, Moments, Variance of a sum, Correlation coefficient, Chebyshev's Inequality. | 11 |
| 2 | **Continuous Probability Distributions:**<br><br>Continuous random variables and their properties, Distribution functions and densities, Normal, Exponential and Gamma densities. | 4 |
| 3 | **Bivariate Distributions:**<br><br>Bivariate distributions and their properties, distribution of sums and quotients, Conditional densities, Bayes' rule. | 5 |
| 4 | **Basic Statistics:**<br><br>Measures of Central tendency, Moments, Skewness and Kurtosis, Probability distributions: Binomial, Poisson and Normal and evaluation of statistical parameters for these three distributions, Correlation and regression – Rank correlation. | 8 |
| 5 | **Applied Statistics:**<br><br>Curve fitting by the method of least squares- fitting of straight lines, second degree parabolas and more general curves. Test of significance: Large sample test for single proportion, difference of proportions, single mean, difference of means, and difference of standard deviations. | 8 |
| 6 | **Small samples:**<br><br>Test for single mean, difference of means and correlation | 4 |

| | | |
|---|---|---|
| | coefficients, test for ratio of variances - Chi-square test for goodness of fit and independence of attributes. | |

**Course Outcomes:**

The students will be able to:

□ Learn the ideas of probability and random variables, various discrete and continuous probability distributions with their properties and their applications in physical and engineering environment.

□ Understand the basic ideas of statistics with different characterisation of a univariate and bivariate data set.

□ Apply statistical tools for analysing data samples and drawing inference on a given data set.

**Learning Resources:**


1. Erwin Kreyszig, Advanced Engineering Mathematics, John Wiley & Sons
2. S. Ross, A First Course in Probability, Pearson Education India
3. W. Feller, An Introduction to Probability Theory and its Applications, Vol. 1, Wiley.
4. John E. Freund, Ronald E. Walpole, Mathematical Statistics, Prentice Hall.
5. B.S. Grewal, Higher Engineering Mathematics, Khanna Publishers.
6. N.G. Das, Statistical Methods (Combined Volume), Tata-McGraw Hill.
7. Reena Garg, Chandrika Prasad, Advanced Engineering Mathematics, Khanna Publishers.

| | |
|---|---|
| **Course Code** : ES-CS201 | **Category :** Engineering Science Courses |
| **Course Title** : Programming for Problem Solving | **Semester** : Second |
| **L-T-P** : **3-0-0** | **Credit:3** |
| **Pre-Requisites:** | |

*Detailed contents*

**Unit 1:** Introduction to Programming **(4 lectures)**

□ Introduction to components of a computer system (disks, memory, processor, where a program is stored and executed, operating system, compilers etc.) - **(1 lecture).**

□ Idea of Algorithm: steps to solve logical and numerical problems. Representation of Algorithm: Flowchart/Pseudocode with examples. **(1 lecture)**

□ From algorithms to programs; source code, variables (with data types) variables and memory locations, Syntax and Logical Errors in compilation, object and executable code- **(2 lectures)**

**Unit 2:** Arithmetic expressions and precedence **(2 lectures) Unit 3:** Conditional Branching and Loops **(6 lectures)**

□ Writing and evaluation of conditionals and consequent branching **(3 lectures)**

□ Iteration and loops **(3 lectures) Unit 4:** Arrays **(6 lectures)**

□ Arrays (1-D, 2-D), Character arrays and Strings

**Unit 5:** Basic Algorithms **(6 lectures)**

□ Searching, Basic Sorting Algorithms (Bubble, Insertion and Selection), Finding roots of equations, notion of order of complexity through example programs (no formal definition required)

**Unit 6:** Function **(5 lectures)**

□ Functions (including using built in libraries), Parameter passing in functions, call by value, Passing arrays to functions: idea of call by reference

**Unit 7:** Recursion **(4 -5 lectures)**

□ Recursion, as a different way of solving problems. Example programs, such as Finding Factorial, Fibonacci series, Ackerman function etc. Quick sort or Merge sort.

**Unit 8:** Structure **(4 lectures)**

□ Structures, Defining structures and Array of Structures

**Unit 9:** Pointers **(2 lectures)**

□ Idea of pointers, Defining pointers, Use of Pointers in self-referential structures, notion of linked list
(no implementation)

**Unit 10:** File handling (only if time is available, otherwise should be done as part of the lab)

**Course Outcomes**

The student will learn

□ To formulate simple algorithms for arithmetic and logical problems.

□ To translate the algorithms to programs (in C language).

□ To test and execute the programs and correct syntax and logical errors.

□ To implement conditional branching, iteration and recursion.

□ To decompose a problem into functions and synthesize a complete program using divide and conquer approach.

□ To use arrays, pointers and structures to formulate algorithms and programs.

□ To apply programming to solve matrix addition and multiplication problems and searching and sorting problems.

□ To apply programming to solve simple numerical method problems, namely rot finding of function, differentiation of function and simple integration.

**Learning Resources:**

1. Byron Gottfried, Schaum's Outline of Programming with C, McGraw-Hill
2. E. Balaguruswamy, Programming in ANSI C, Tata McGraw-Hill
3. Brian W. Kernighan and Dennis M. Ritchie, The C Programming Language, Prentice Hall of India
4. R. S. Salaria, Computer Concepts and Programming in C, Khanna Publishers

| | |
|---|---|
| **Course Code** : HSMC 201 | **Category :** Humanities and Social Sciences including Management courses |
| **Course Title** : English | **Semester** : Second |
| **L-T-P : 2-0-0** | **Credit:2** |
| **Pre-Requisites:** | |

*Detailed contents*

**1. Vocabulary Building**

1.1 The concept of Word Formation

1.2 Root words from foreign languages and their use in English

1.3 Acquaintance with prefixes and suffixes from foreign languages in English to form derivatives.

1.4 Synonyms, antonyms, and standard abbreviations.

**2. Basic Writing Skills**

2.1 Sentence Structures
2.2 Use of phrases and clauses in sentences
2.3 Importance of proper punctuation
2.4 Creating coherence
2.5 Organizing principles of paragraphs in documents
2.6 Techniques for writing precisely

**3. Identifying Common Errors in Writing**
3.1 Subject-verb agreement
3.2 Noun-pronoun agreement
3.3 Misplaced modifiers
3.4 Articles
3.5 Prepositions
3.6 Redundancies
3.7 Clichés
**4. Nature and Style of sensible Writing**
4.1 Describing
4.2 Defining
4.3 Classifying
4.4 Providing examples or evidence
4.5 Writing introduction and conclusion

**5. Writing Practices**
5.1 Comprehension
5.2 Précis Writing
5.3 Essay Writing

**6. Oral Communication**

(This unit involves interactive practice sessions in Language Lab)

□ Listening Comprehension
□ Pronunciation, Intonation, Stress and Rhythm
□ Common Everyday Situations: Conversations and Dialogues
□ Communication at Workplace
□ Interviews
□ Formal Presentations

**Learning Resources:**

(i) Practical English Usage. Michael Swan. OUP. 1995.

(ii) Remedial English Grammar. F.T. Wood. Macmillan.2007
(iii) On Writing Well. William Zinsser. Harper Resource Book. 2001
(iv) Study Writing. Liz Hamp-Lyons and Ben Heasly. Cambridge University Press. 2006.
(v) Communication Skills. Sanjay Kumar and PushpLata. Oxford University Press. 2011.
(vi) Exercises in Spoken English. Parts. I-III. CIEFL, Hyderabad. Oxford University Press
(vii) Kulbushan Kumar, R S Salaria, Effective Communication Skills, Khanna Publishing House, Delhi.

**Course Outcomes**

The student will acquire basic proficiency in English including reading and listening comprehension, writing and speaking skills.

| | |
|---|---|
| **Course Code** : BS-PH291 | **Category :** Basic Science course |
| **Course Title** : Physics-I Lab | **Semester** : First/ Second |
| **L-T-P** : **0-0-4** | **Credit:2** |
| **Pre-Requisites:** | |

**Choose 10 experiments including at least one from Optics, Electricity and Magnetism and Quantum Mechanics and at least a total of six from these three groups.**

**Experiments in Optics**

1. Determination of dispersive power of the material of a prism
2. Determination of wavelength of a monochromatic light by Newton's ring
3. Determination of wavelength of a monochromatic light by Fresnel's bi-prism
4. Determination of wavelength of the given laser source by diffraction method

**Electricity & Magnetism experiments**

1. Determination of thermo electric power of a given thermocouple.
2. Determination of specific charge (e/m) of electron by J.J. Thompson's method.
3. Determination of dielectric constant of a given dielectric material.
4. Determination of Hall coefficient of a semiconductor by four probe method.
5. To study current voltage characteristics, load response, areal characteristic and spectral response of a photovoltaic solar cell.

6. Determination of resistance of ballistic galvanometer by half deflection method and study of variation of logarithmic decrement with series resistance.
7. Determination of unknown resistance using Carey Foster's bridge
8. Study of Transient Response in LR, RC and LCR circuits using expeyes
9. Generating sound from electrical energy using expeyes

**Experiments in Quantum Physics**

1. Determination of Stefan-Boltzmann constant.
2. Determination of Planck constant using photocell.
3. Determination of Lande-g factor using Electron spin resonance spectrometer.
4. Determination of Rydberg constant by studying Hydrogen spectrum.
5. Determination of Band gap of semiconductor.
6. To study current voltage characteristics, load response, areal characteristic and spectral response of a photovoltaic solar cell.

**Miscellaneous experiments**

1. Determination of Young's modulus of elasticity of the material of a bar by the method of flexure
2. Determination of bending moment and shear force of a rectangular beam of uniform cross-section
3. Determination of modulus of rigidity of the material of a rod by static method
4. Determination of rigidity modulus of the material of a wire by dynamic method
5. To determine the moment of inertia of a body about an axis passing through its centre of gravity and to determine the modulus of rigidity of the material of the suspended wire
6. Determination of coefficient of viscosity by Poiseulle's capillary flow method

| | |
|---|---|
| **Course Code** : ES-CS291 | **Category :** Engineering Science Courses |
| **Course Title** : Programming for Problem Solving | **Semester** : Second |
| **L-T-P** : **0-0-4** | **Credit:2** |
| **Pre-Requisites:** | |

*The laboratory should be preceded or followed by a tutorial to explain the approach or algorithm to be implemented for the problem given.*

**Tutorial 1:** Problem solving using computers:
**Lab1:** Familiarization with programming environment

**Tutorial 2:** Variable types and type conversions:
**Lab 2:** Simple computational problems using arithmetic expressions
**Tutorial 3:** Branching and logical expressions:
**Lab 3**: Problems involving if-then-else structures
**Tutorial 4:** Loops, while and for loops:
**Lab 4:** Iterative problems e.g., sum of series
**Tutorial 5:** 1D Arrays: searching, sorting:
**Lab 5:** 1D Array manipulation
**Tutorial 6:** 2D arrays and Strings
**Lab 6:** Matrix problems, String operations
**Tutorial 7:** Functions, call by value:
**Lab 7:** Simple functions
**Tutorial 8 &9:** Numerical methods (Root finding, numerical differentiation, numerical integration):
**Lab 8 and 9:** Programming for solving Numerical methods problems
**Tutorial 10:** Recursion, structure of recursive calls
**Lab 10:** Recursive functions
**Tutorial 11:** Pointers, structures and dynamic memory allocation
**Lab 11:** Pointers and structures
**Tutorial 12:** File handling:
**Lab 12:** File operations

**Laboratory Outcomes**

□ To formulate the algorithms for simple problems
□ To translate given algorithms to a working and correct program
□ To be able to correct syntax errors as reported by the compilers
□ To be able to identify and correct logical errors encountered at run time
□ To be able to write iterative as well as recursive programs
□ To be able to represent data in arrays, strings and structures and manipulate them through a program
□ To be able to declare pointers of different types and use them in defining self-referential structures.
□ To be able to create, read and write to and from simple text files.

| | |
|---|---|
| **Course Code** : ES-ME 291 | **Category :** Engineering Science Courses |
| **Course Title** : Workshop/ Manufacturing Practices | **Semester** : Second |
| **L-T-P** : **1-0-4** | **Credit:3** |
| **Pre-Requisites:** | |

**(i) Lectures & videos:**

Detailed contents:

1. Manufacturing Methods- casting, forming, machining, joining, advanced manufacturing methods
2. CNC machining, Additive manufacturing
3. Fitting operations & power tools
4. Electrical &Electronics
5. Carpentry
6. Plastic moulding, glass cutting
7. Metal casting
8. Welding (arc welding & gas welding), brazing

**(ii) Workshop Practice:**

□ **Machine shop (8 hours)**

*Typical jobs that may be made in this practice module:*

□ To make a pin from a mild steel rod in a lathe.

□ To make rectangular and vee slot in a block of cast iron or mild steel in a shaping and / or milling machine

□ **Fitting shop (8 hours)**

*Typical jobs that may be made in this practice module:*

□ To make a Gauge from MS plate.

□ **Carpentry (8 hours)**

*Typical jobs that may be made in this practice module:*

□ To make wooden joints and/or a pattern or like.

□ **Welding shop (8 hours (Arc welding 4 hrs + gas welding 4 hrs))**

*Typical jobs that may be made in this practice module:*

□ ARC WELDING (4 hours): To join two thick (approx. 6mm) MS plates by manual metal arc welding.

□ GAS WELDING (4 hours): To join two thin mild steel plates or sheets by gas welding.

□ **Casting (8 hours)**

*Typical jobs that may be made in this practice module:*

□ One/two green sand moulds to prepare, and a casting be demonstrated.

□ **Smithy (4 hours) ~ 4 hours**

*Typical jobs that may be made in this practice module:*

□ A simple job of making a square rod from a round bar or like.

□ **Plastic moulding & Glass cutting (4 hours)**

*Typical jobs that may be made in this practice module:*

□ For plastic moulding, making at least one simple plastic component should be made.

□ For glass cutting, three rectangular glass pieces may be cut to make a kaleidoscope using a black colour diamond cutter, or similar other components may be made.

□ **Electrical & Electronics (8 hours)**

□ Familiarization with LT switchgear elements, making its sketches and noting down its specification. Kitkat fuse, Glass cartridge fuse, Plastic fuse holders (optional), Iron clad isolators, MCB style isolators, Single phase MCB, Single-phase wire, wiring cable.

□ Demonstration of domestic wiring involving two MCB, two piano key switches, one incandescent lamp, one LED lamp and plug point.

□ Simple wiring exercise to be executed to understand the basic electrical circuit.

□ Simple soldering exercises to be executed to understand the basic process of soldering.

□Fabrication of a single-phase full wave rectifier with a step down transformer using four diodes and electrolytic capacitor and to find its volt-ampere characteristics to understand basic electronic circuit fabrication.

**Examinations could involve the actual fabrication of simple components, utilizing one or more of the techniques covered above.**

**Laboratory Outcomes**

□ Upon completion of this laboratory course, students will be able to fabricate components with their own hands.

□ They will also get practical knowledge of the dimensional accuracies and dimensional tolerances possible with different manufacturing processes.

□ By assembling different components, they will be able to produce small devices of their interest.

**Learning Resources:**

1. Hajra Choudhury S.K., Hajra Choudhury A.K. and Nirjhar Roy S.K., "Elements of Workshop
Technology", Vol. I 2008 and Vol. II 2010, Media promoters and publishers private limited, Mumbai.
2. Kalpakjian S. and Steven S. Schmid, "Manufacturing Engineering and Technology", 4th edition,
Pearson Education India Edition, 2002.
3. Gowri P. Hariharan and A. Suresh Babu,"Manufacturing Technology – I" Pearson Education, 2008.
4. Roy A. Lindberg, "Processes and Materials of Manufacture", 4th edition, Prentice Hall India, 1998.
5. Rao P.N., "Manufacturing Technology", Vol. I and Vol. II, Tata McGrawHill House, 2017.

| | |
|---|---|
| **Course Code** : HSMC 291 | **Category :** Humanities and Social Sciences including Management courses |
| **Course Title** : Language Lab | **Semester** : Second |
| **L-T-P** : **0-0-2** | **Credit:1** |
| **Pre-Requisites:** | |

1) Honing 'Listening Skill' and its sub skills through Language Lab Audio device; 3P

2) Honing 'Speaking Skill' and its sub skills 2P

3)Helping them master Linguistic/Paralinguistic features (Pronunciation/Phonetics/Voice modulation/ Stress/ Intonation/ Pitch &Accent) of connected speech 2P

4) Honing 'Conversation Skill' using Language Lab Audio –Visual input; Conversational Practice Sessions (Face to Face / via Telephone, Mobile phone &Role Play Mode)2P

5) Introducing ‘Group Discussion’ through audio –Visual input and acquainting them with key strategies for success 2P
6) G D Practice Sessions for helping them internalize basic Principles (turn taking, creative intervention, by using correct body language, courtesies & other soft skills) of GD 4P
7) Honing ‘Reading Skills’ and its sub skills using Visual / Graphics/ Diagrams /Chart Display/Technical/Non Technical Passages Learning Global/Contextual / Inferential Comprehension; 2P
8) Honing ‘Writing Skill’ and its sub skills by using Language Lab Audio –Visual input; Practice Sessions 2P
Course Outcomes
□ The student will acquire basic proficiency in English including reading and listening comprehension, writing and speaking skills.

| Sl. | Chapter | Title |
|---|---|---|
| | | General, Course Structure, Theme & Semester wise Credit |
| **1** | **1** | Distribution |
| **2** | | Detailed 4-YEAR Curriculum Contents |
| | **(i)** | **Professional Core Courses** |
| | | PCC-CS301: Principles of Programming Language |
| | | PCC-CS302: Data structure & Algorithm |
| | | PCC-CS303: IT Workshop –(Python/R/MATLAB/Scilab) |
| | | PCC-CS401: Discrete Mathematics |
| | | PCC-CS402: Computer Organization & Architecture |
| | | PCC-CS403: Operating Systems |
| | | PCC-CS-404: Design and Analysis of Algorithms |
| | | PCC-CS501: Software Engineering |
| | | PCC-CS502: Database Management Systems |
| | | PCC-CS503: Formal Language & Automata Theory |
| | | PCC-CS504: Object Oriented Programming |
| | | PCC-CS601: Compiler Design |
| | | PCC-CS602: Computer Networks |
| | | PCC-CS391: Principles of Programming Language |
| | | PCC-CS392: Data structure & Algorithm |
| | | PCC-CS393: IT Workshop –(Python/R/MATLAB/Scilab) |
| | | PCC-CS492:Computer Organization & Architecture |
| | | PCC-CS493: Operating Systems |
| | | PCC-CS494: Design and Analysis of Algorithms |
| | | PCC-CS592: Database Management Systems |
| | | PCC-CS594: Object Oriented Programming |
| | | PCC-CS691: Compiler Design |
| | | PCC-CS692: Computer Networks |
| | **(ii)** | **Professional Elective Courses** |
| | | Additional Courses for B.Tech (Hons.) |
| **3** | **Appendix-A** | **A Guide to Induction Program** |
| | | Common courses (Physics, Chemistry, Biology & |
| | | Mathematics) |
| **4** | | MC: Model Curriculum for Mandatory Non-credit courses |
| | | HSMC: Model Curriculum for courses in Humanities and |
| **5** | | Social Sciences including Management |
| **6** | | Virtual Laboratories for various disciplines |

## Chapter -1
## General, Course structure & Theme
## &
## Semester-wise credit distribution

### A. Definition of Credit

| | |
|---|---|
| 1. Hr. Lecture (L) per week | 1 credit |
| 2. Hr. Tutorial (T) per week | 1 credit |
| 3. Hr. Practical (P) per week | 0.5 credit |
| 4. 2 Hours Practical(Lab)/week | 1 credit |

**B. Range of credits -**A range of credits from 150 to 160 for a student to be eligible to get Under Graduate degree in Engineering. A student will be eligible to get Under Graduate degree with Honours or additional Minor Engineering, if he/she completes an additional 20 credits. These could be acquired through MOOCs.

### C. Structure of Undergraduate Engineering program :

| Sl. No. | Course | Credit Breakup for CSE students |
|---|---|---|
| 1 | Humanities and Social Sciences including Management courses | 12 |
| 2 | Basic Science courses | 25 |
| 3 | Engineering Science courses including workshop, drawing, basics of electrical/mechanical/computer etc | 21 |
| 4 | Professional core courses | 57 |
| 5 | Professional Elective courses relevant to chosen specialization/branch | 18 |
| 6 | Open subjects – Electives from other technical and /or emerging subjects | 12 |
| 7 | Project work, seminar and internship in industry or elsewhere | 15 |
| 8 | Mandatory Courses [Environmental Sciences, Induction Program, Indian Constitution, Essence of Indian Traditional Knowledge] | (non-credit) |
| | Total | 160* |

**Minor variation is allowed as per need of the respective disciplines.*

**D. Credit distribution in the First year of Undergraduate Engineering program :**

| Course | Lecture | Tutorial | Laboratory/ Practical | Total credits |
|---|---|---|---|---|
| Chemistry-I | 3 | 1 | 2 | 6 |
| Physics-I | 3 | 1 | 2 | 6 |
| Mathematics-I | 3 | 1 | 0 | 4 |
| Mathematics-II | 3 | 1 | 0 | 4 |
| Programming for Problem Solving | 3 | 0 | 4 | 5 |
| English | 2 | 0 | 2 | 3 |
| Engineering Graphics &Design | 1 | 0 | 4 | 3 |
| Workshop/Manufacturing Practices | 1 | 0 | 4 | 3 |
| Basic Electrical & Electronics Engineering | 3 | 1 | 2 | 5 |
| *Biology | 2 | 1 | 0 | 3 |
| *Mathematics-III | 3 | 1 | 0 | 4 |

***These courses may be offered preferably in the later semesters***

***E. Course code and definition:***

| *Course code* | *Definitions* |
|---|---|
| BS | Basic Science Courses |
| ES | Engineering Science Courses |
| HSMC | Humanities and Social Sciences including Management courses |
| PCC-CS | Professional core courses |
| PEC –CS | Professional Elective courses |
| OEC-CS | Open Elective courses |
| LC | Laboratory course |
| MC | Mandatory courses |
| SI | Summer Industry Internship |
| PROJ-CS | Project |

## HUMANITIES AND SOCIAL SCIENCES INCLUDING MANAGEMENT COURSES

| Sl. No | Code No. | Course Title | Hours per week | | | Total Credits | Semester |
|---|---|---|---|---|---|---|---|
| | | | Lecture | Tutorial | Practical | | |
| 1 | HSMC 201 | English | 2 | 0 | 2 | 3 | 2 |
| 2 | HSMC 301 | Humanities – I | 3 | 0 | 0 | 3 | 3 |
| 3 | HSMC 401 | Management-I (Organizational Behaviour)/ Finance & Accounting | 3 | 0 | 0 | 3 | 4 |
| 4 | HSMC 501 | Humanities – II | 3 | 0 | 0 | 3 | 5 |
| **Total Credits:** | | | | | | **12** | |

## BASIC SCIENCE COURSE [BS]

| Sl. No | Code No. | Course Title | Hours per week | | | Total Credits | Semester |
|---|---|---|---|---|---|---|---|
| | | | Lecture | Tutorial | Practical | | |
| 1 | BS-CH101 | Chemistry-I | 3 | 1 | 4 | 6 | 1 |
| 2 | BS-M101 | Mathematics-I (Calculus and Linear Algebra) | 3 | 1 | 0 | 4 | 1 |
| 3 | BS-PH201 | Physics-I | 3 | 1 | 4 | 6 | 2 |
| 4 | BS-M201 | Mathematics-II (Probability and Statistics) | 3 | 1 | 0 | 4 | 2 |
| 5 | BS-M301 | Mathematics-III (Differential Calculus) | 2 | 0 | 0 | 2 | 3 |
| 6 | BS-B701 | Biology | 2 | 1 | 0 | 3 | 7 |
| **Total Credits:** | | | | | | **25** | |

## ENGINEERING SCIENCE COURSE [ES]

| Sl. No | Code No. | Course Title | Hours per week | | | Total Credits | Semester |
|---|---|---|---|---|---|---|---|
| | | | Lecture | Tutorial | Practical | | |
| 1 | ES-EE101 | Basic Electrical & Electronics Engineering | 4 | 0 | 2 | 5 | 1 |
| 2 | ES-ME101 | Engineering Graphics &Design | 1 | 0 | 4 | 3 | 1 |
| 3 | ES-CS201 | Programming for Problem Solving | 3 | 0 | 4 | 5 | 2 |
| 4 | ES-ME201 | Workshop/Manufacturing Practices | 1 | 0 | 4 | 3 | 2 |
| 5 | ES-CS301 | Digital Electronics | 3 | 0 | 4 | 5 | 3 |
| **Total Credits:** | | | | | | **21** | |

## PROFESSIONAL CORE COURSES [PCC]

| Sl. No | Code No. | Course Title | Hours per week | | | Total Credits | Semester |
|---|---|---|---|---|---|---|---|
| | | | Lecture | Tutorial | Practical | | |
| 1 | PCC-CS301 | Principles of Programming Language | 3 | 0 | 4 | 5 | 3 |
| 2 | PCC-CS302 | Data Structure & Algorithms | 3 | 0 | 4 | 5 | 3 |
| 3 | PCC-CS303 | IT Workshop – (Python/R/ MATLAB/Scilab) | 1 | 0 | 4 | 3 | 3 |
| 4 | PCC-CS401 | Discrete Mathematics | 3 | 1 | 0 | 4 | 4 |
| 5 | PCC-CS402 | Computer Organization and Architecture | 3 | 0 | 4 | 5 | 4 |
| 6 | PCC-CS403 | Operating Systems | 3 | 0 | 4 | 5 | 4 |
| 7 | PCC-CS404 | Design and Analysis of Algorithms | 3 | 0 | 4 | 5 | 4 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 8 | PCC-CS501 | Software Engineering | 3 | 0 | 0 | 3 | 5 |
| 9 | PCC-CS502 | Database Management Systems | 3 | 0 | 4 | 5 | 5 |
| 10 | PCC-CS503 | Formal Language, Automats and Complier | 3 | 0 | 0 | 3 | 5 |
| 11 | PCC-CS504 | Object Oriented Programming | 2 | 0 | 4 | 4 | 5 |
| 12 | PCC-CS601 | Compiler Design | 3 | 0 | 4 | 5 | 6 |
| 13 | PCC-CS602 | Computer Networks | 3 | 0 | 4 | 5 | 6 |
| **Total Credits** | | | | | | **57** | |

**PROFESSIONAL ELECTIVE COURSES [PEC]**

| **Sl. No** | **Code No.** | **Course Title** | **Hours per week** | | | **Total** Credits | **Semester** |
|---|---|---|---|---|---|---|---|
| | | | Lecture | Tutorial | Practical | | |
| 1 | PEC-CS501 | Elective – I | 3 | 0 | 0 | 3 | 5 |
| 2 | PEC-CS601 | Elective-II | 3 | 0 | 0 | 3 | 6 |
| 3 | PEC-CS602 | Elective-III | 3 | 0 | 0 | 3 | 6 |
| 4 | PEC-CS701 | Elective-IV | 3 | 0 | 0 | 3 | 7 |
| 5 | PEC-CS702 | Elective-V | 3 | 0 | 0 | 3 | 7 |
| 6 | PEC-CS801 | Elective-VI | 3 | 0 | 0 | 3 | 8 |
| **Total Credits** | | | | | | **18** | |

**OPEN ELECTIVE COURSES [OEC]**

| **Sl. No** | **Code No.** | **Course Title** | **Hours per week** | | | **Total** Credits | **Semester** |
|---|---|---|---|---|---|---|---|
| | | | Lecture | Tutorial | Practical | | |
| 1 | OEC-CS601 | Open Elective-I | 3 | 0 | 0 | 3 | 6 |
| 2 | OEC-CS701 | Open Elective-II | 3 | 0 | 0 | 3 | 7 |
| 3 | OEC-CS801 | Open Elective-III | 3 | 0 | 0 | 3 | 8 |
| 4 | OEC-CS802 | Open Elective-IV | 3 | 0 | 0 | 3 | 8 |
| **Total Credits** | | | | | | **12** | |

# 4 year Curriculum structure

**Undergraduate Degree in Engineering & Technology**

**Branch / course : Computer Science and Engineering**

**Total credits (4 year course): 160**

## I. Induction Program (Please refer Appendix-A for guidelines)

| | |
|---|---|
| **Induction program (mandatory)** | **3 weeks duration**<br>**(**Please refer Appendix-A for guidelines & also details<br>available in the curriculum of<br>Mandatory courses) |
| Induction program for students to be offered right at the start of the first year. | Physical activity<br>Creative Arts<br>Universal Human Values<br>Literary<br>Proficiency Modules<br>Lectures by Eminent People<br>Visits to local Areas<br>Familiarization to Dept./Branch & Innovations |

# Semester-wise structure of curriculum

**[L= Lecture, T = Tutorials, P = Practical & C = Credits]**

## Semester I (First year) Curriculum
## Computer Science Engineering

### First Year First Semester

| Sl. No | Category | Subject Code | Subject Name | Total Number of contact hours | | | Credits |
|---|---|---|---|---|---|---|---|
| | | | | L | T | P | |
| 1 | Basic Science course | BS-CH101 | Chemistry-I | 3 | 1 | 0 | 4 |
| 2 | Basic Science course | BS-M101 | Mathematics –I (Calculus & Linear Algebra) | 3 | 1 | 0 | 4 |
| 3 | Engineering Science Courses | ES-EE101 | Basic Electrical& Electronics Engineering | 4 | 0 | 0 | 4 |
| 4 | Engineering Science Courses | ES-ME 101 | Engineering Graphics & Design | 1 | 0 | 0 | 1 |
| 5 | Basic Science course | BS-CH191 | Chemistry-I Lab | 0 | 0 | 4 | 2 |
| 6 | Engineering Science Courses | ES-EE191 | Basic Electrical& Electronics Engineering Lab | 0 | 0 | 2 | 1 |
| 7 | Engineering Science Courses | ES-ME191 | Engineering Graphics & Design | 0 | 0 | 4 | 2 |
| **Total Credits** | | | | | | | **18** |

## Semester II (First year) Curriculum
## Computer Science Engineering

| Sl. No | Category | Subject Code | Subject Name | Total Number of contact hours | | | Credits |
|---|---|---|---|---|---|---|---|
| | | | | L | T | P | |
| **1** | Basic Science courses | BS-PH201 | Physics-I | 3 | 1 | 0 | 4 |
| **2** | Basic Science courses | BS-M201 | Mathematics-II (Probability and Statistics) | 3 | 1 | 0 | 4 |
| **3** | Engineering Science Courses | ES-CS201 | Programming for Problem Solving | 3 | 0 | 0 | 3 |
| **4.** | Engineering Science Courses | ES-ME201 | Workshop/Manufacturing Practices | 1 | 0 | 0 | 1 |
| **5** | Humanities and Social Sciences including Management courses | HSMC 201 | English | 2 | 0 | 0 | 2 |
| **6** | Basic Science courses | BS-PH291 | Physics- I Lab | 0 | 0 | 4 | 2 |
| **7** | Engineering Science Courses | ES-CS291 | Programming for Problem Solving Lab | 0 | 0 | 4 | 2 |
| **8** | Engineering Science Courses | ES-ME291 | Workshop/Manufacturing Practices | 0 | 0 | 4 | 2 |
| **9** | Humanities and Social Sciences including Management courses | HSMC 291 | Language Lab | 0 | 0 | 2 | 1 |
| **Total of Credits** | | | | | | | **21** |

# Semester III (Second year) Curriculum
# Computer Science Engineering

| Sl. No | Category | Subject Code | Subject Name | Total Number of contact hours | | | Credits |
|---|---|---|---|---|---|---|---|
| | | | | L | T | P | |
| 1 | Professional Core Courses | PCC-CS301 | Principles of Programming Language | 3 | 0 | 0 | 3 |
| 2 | Professional Core Courses | PCC-CS302 | Data structure & Algorithms | 3 | 0 | 0 | 3 |
| 3 | Engineering Science Courses | ES-CS301 | Digital Electronics | 3 | 0 | 0 | 3 |
| 4 | Professional Core Courses | PCC-CS303 | IT Workshop (Python/ R/ Sci Lab/ MATLAB) | 1 | 0 | 0 | 1 |
| 5 | Basic Science courses | BS-M301 | Mathematics-III (Differential Calculus) | 2 | 0 | 0 | 2 |
| 6 | Humanities & Social Sciences including Management Courses | HSMC 301 | Humanities-I | 3 | 0 | 0 | 3 |
| 7 | Professional Core Course | PCC-CS391 | Principles of Programming Language Lab | 0 | 0 | 4 | 2 |
| 8 | Professional Core Courses | PCC-CS392 | Data structure & Algorithms Lab | 0 | 0 | 4 | 2 |
| 9 | Engineering Science Courses | ES-CS391 | Digital Electronics Lab | 0 | 0 | 4 | 2 |
| 10 | Professional Core Courses | PCC-CS393 | IT Workshop (Python/R/ Sci Lab/ MATLAB) Lab | 0 | 0 | 4 | 2 |
| **Total Credits** | | | | | | | **23** |

## Semester IV (Second year) Curriculum
## Computer Science Engineering

| Sl. No | Category | Subject Code | Subject Name | Total Number of contact hours | | | Credits |
|---|---|---|---|---|---|---|---|
| | | | | L | T | P | |
| **1** | Professional Core Course | PCC-CS401 | Discrete Mathematics | 3 | 1 | 0 | 4 |
| **2** | Professional Core Courses | PCC-CS402 | Computer Organization & Architecture | 3 | 0 | 0 | 3 |
| **3** | Professional Core Courses | PCC-CS403 | Operating Systems | 3 | 0 | 0 | 3 |
| **4** | Professional Core Courses | PCC-CS404 | Design & Analysis of Algorithms | 3 | 0 | 0 | 3 |
| **5** | Humanities & Social Sciences including Management Courses | HSMC-401 | Management-I (Organizational Behaviour/ Finance &Accounting) | 3 | 0 | 0 | 3 |
| **6** | Mandatory Courses | MC-401 | Environmental Sciences | 1 | 0 | 0 | 0 |
| **7** | Professional Core Courses | PCC-CS492 | Computer Organization & Architecture Lab | 0 | 0 | 4 | 2 |
| **8** | Professional Core Courses | PCC-CS493 | Operating Systems Lab | 0 | 0 | 4 | 2 |
| **9** | Professional Core Courses | PCC-CS494 | Design & Analysis of Algorithms Lab | 0 | 0 | 4 | 2 |
| **Total Credits** | | | | | | | **22** |

# Semester V (Third year) Curriculum
# Computer Science Engineering

| Sl. No | Category | Subject Code | Subject Name | Total Number of contact hours | | | Credits |
|---|---|---|---|---|---|---|---|
| | | | | L | T | P | |
| 1 | Professional Core Course | PCC-CS501 | Software Engineering | 3 | 0 | 0 | 3 |
| 2 | Professional Core Course | PCC-CS502 | Database Management Systems | 3 | 0 | 0 | 3 |
| 3 | Professional Core Courses | PCC-CS503 | Formal Language &Automata Theory | 3 | 0 | 0 | 3 |
| 4 | Professional Core Courses | PCC-CS504 | Object Oriented Programming | 2 | 0 | 0 | 2 |
| 5 | Humanities & Social Sciences including Management Courses | HSMC-501 | Humanities-II | 3 | 0 | 0 | 3 |
| 6 | Professional Elective Courses | PEC-CS501 A/B/C/D/E | A. Graph Theory<br>B. Signals & Systems<br>C. Artificial Intelligence<br>D. Image Processing<br>E. Soft Computing | 3 | 0 | 0 | 3 |
| 10 | Mandatory Course | MC-501 | Constitution of India | 1 | 0 | 0 | 0 |
| 11 | Professional Core Course | PCC-CS592 | Database Management Systems Lab | 0 | 0 | 4 | 2 |
| 12 | Professional Core Courses | PCC-CS594 | Object Oriented Programming Lab | 0 | 0 | 4 | 2 |
| **Total Credits** | | | | | | | **21** |

# Semester VI (Third year) Curriculum
# Computer Science Engineering

| Sl. No | Category | Subject Code | Subject Name | Total Number of contact hours | | | Credits |
|---|---|---|---|---|---|---|---|
| | | | | L | T | P | |
| 1 | Professional Core Course | PCC-CS601 | Compiler Design | 3 | 0 | 0 | 3 |
| 2 | Professional Core Course | PCC-CS602 | Computer Networks | 3 | 0 | 0 | 3 |
| 3 | Professional Elective Courses | PEC-CS601 A/B/C/D/E | A. Advanced Algorithms<br>B. Distributed Database<br>C. Real Time Systems<br>D. Information Retrieval<br>E. Advanced Computer Architecture | 3 | 0 | 0 | 3 |
| 4 | Professional Elective Courses | PEC-CS602 A/B/C/D/E | A. Computer Graphics<br>B. Optimization Techniques<br>C. Information Theory & Coding<br>D. Parallel & Distributed Algorithm<br>E. Internet of Things | 3 | 0 | 0 | 3 |
| 5 | Open Elective Courses | OEC-CS601 | Soft Skills and Interpersonal Communication | 3 | 0 | 0 | 3 |
| 6 | Project | PROJ-CS691 | Project-I | 0 | 0 | 6 | 3 |
| 7 | Professional Core Course | PCC- CS691 | Compiler Design Lab | 0 | 0 | 4 | 2 |
| 8 | Professional Core Course | PCC-CS692 | Computer Networks Lab | 0 | 0 | 4 | 2 |
| **Total Credits** | | | | | | | **22** |

## Semester VII (Fourth year) Curriculum
## Computer Science Engineering

| Sl. No | Category | Subject Code | Subject Name | Total Number of contact hours | | | Credits |
|---|---|---|---|---|---|---|---|
| | | | | L | T | P | |
| 1 | Professional Elective Courses | PEC-CS701 A/B/C/D/E/F | A. Adhoc and Sensor Networks<br>B. Machine Learning<br>C. Neural Networks & Deep Learning<br>D. Advanced Operating System<br>E. Computational Geometry<br>F. Web & Internet | 3 | 0 | 0 | 3 |
| 2 | Professional Elective Courses | PEC-CS702 A/B/C/D/E/F | A. Speech & Natural Language Processing<br>B. Human Computer Interaction<br>C. VLSI Design<br>D. Data Analytics<br>E. Theory of Computation<br>F. System Software & Administration | 3 | 0 | 0 | 3 |
| 3 | Open Elective Courses | OEC-CS701 A/B | A. Human Resource Development and Organizational Behaviour<br>B. Indian Music System | 3 | 0 | 0 | 3 |
| 4 | Basic Science course | BS-B701 | Biology | 2 | 1 | 0 | 3 |
| 5 | Project | PROJ-CS791 | Project-II | 0 | 0 | 12 | 6 |
| **Total Credits** | | | | | | | **18** |

# Semester VIII (Fourth year) Curriculum
# Computer Science Engineering
# [Summer Industry Internship]

| Sl. No | Category | Subject Code | Subject Name | Total Number of contact hours | | | Credits |
|---|---|---|---|---|---|---|---|
| | | | | L | T | P | |
| 1 | Professional Elective Courses | PEC-CS801 A/B/C/D/E /F | A. Cyber Security<br>B. Quantum Computing<br>C. Cryptography & Network Security<br>D. Cloud Computing<br>E. Embedded Systems<br>F. Data Mining | 3 | 0 | 0 | 3 |
| 2 | Open Elective Courses | OEC-CS801 | Cyber Law and Ethics | 3 | 0 | 0 | 3 |
| 4 | Open Elective Courses | OEC-CS802 | Economic Policies in India | 3 | 0 | 0 | 3 |
| 5 | Project | PROJ-CS891 | Project-III | 0 | 0 | 12 | 6 |
| **Total Credits** | | | | | | | **15** |
| **Total Credit: Sem I+Sem II+Sem III+Sem IV+Sem V+Sem VI+Sem VII+Sem VIII** | | | | | | | **160** |

# CHAPTER 2

## DETAILED 4-YEAR CURRICULUM CONTENTS

## Undergraduate Degree in Engineering & Technology

## Branch/Course: COMPUTER SCIENCE AND ENGINEERING

## Second year (Third semester onwards)

**COMPUTER SCIENCE AND ENGINEERING**

# Semester-III (2nd Year)

| | |
|---|---|
| Course Code: PCC-CS301 | Category: Professional Core Courses |
| **Course Title: Principles of Programming Languages** | Semester: 3rd Semester |
| L-T-P: 3-0-0 | Credit: 3 |
| **Teaching Scheme**<br>Theory: 3hrs/week<br>Tutorial: 0 hr/week | **Examination Scheme**<br>Class Tests & Assignments: 25 marks<br>Attendance: 5 marks<br>End Semester Exam: 70 marks |
| | |
| | |

## ***Detailed Syllabus***

**Course Objectives:**

1. Provides students exposure to examples of important programming languages and paradigms such as LISP/Scheme, ML/Haskell, Prolog, C++/Python/Ruby, etc.

2. The languages are used to illustrate programming language constructs such as binding, binding times, data types and implementation, operations (assignment data-type creation, pattern matching), data control, storage management, parameter passing, and operating environment.

3. The suitability of these various languages for particular programming tasks.

4. Main goal for this course is to enable the students to understand enough about the fundamental principles of programming languages so that they are able to easily and quickly pick up any new programming language. Their understanding of any CS topic should not be tied to any particular programming language.

**Pre-requisites:**

A first course in programming (using any structured programming language such as C) such as PCC-CS-201 and PCC-CS-291.

**Course Outcomes:**

On completion of this course, students should have

1. Knowledge of, and ability to use, language features used in current programming languages

2. An ability to program in different language paradigms and evaluate their relative benefits
3. An understanding of the key concepts in the implementation of common features of programming languages

**Syllabus:**

Principles of Language Design, Programming language spectrum, Why study Programming Languages, Overview of Compilation, Interpretation, and Programming Environments [~2 hrs]

Specifications of Language Syntax [~2 hrs]

Fundamentals of Procedural and Object Oriented Languages [~6 hrs]

Fundamentals of Functional Programming [~9 hrs]

Fundamentals of Logic Programming [~6 hrs]

Programming Language Semantics, Values, Bindings, Types [~6 hrs]

Programming Language Constructs, Expressions, Statements, Procedures and Environments, Parameter passing [~5 hrs]

**Textbook:**

"Programming Language Pragmatics", 4e, by *Michael Scott*. Morgan Kaufmann Publishers

"Programming Languages: Principles and Paradigms", 2010th edition, by *Maurizio Gabbrielli, Simone Martini*. Springer

**Reference:**

"Concepts of Programming Languages", 11e, by *Robert W. Sebesta.* Pearson India

"Types and Programming Languages", *Benjamin C. Pierce*. MIT Press

| YEAR- 2ND | SEMESTER- III | PROFESSIONAL CORE COURSES | |
|---|---|---|---|
| PCC-CS302 | Data Structure & Algorithms | 3L:0T: 0P | 3 credits |
| Pre-requisites | ES-CS201 | | |

**Objectives of the course:**

1. To impart the basic concepts of data structures and algorithms.
2. To understand concepts about searching and sorting techniques
3. To understand basic concepts about stacks, queues, lists, trees and graphs.
4. To enable them to write algorithms for solving problems with the help of fundamental data structures

**Detailed contents:**

**Module 1: [ 4L]**

**Introduction:** Basic Terminologies: Elementary Data Organizations, Data Structure Operations: insertion, deletion, traversal etc.; Analysis of an Algorithm, Asymptotic Notations, Time-Space trade off.

**Array :** Operations , Different representations – row major, column major.

Sparse matrix - its implementation and usage. Array representation of polynomials.

**Module 2: [12 L]**

**Stacks and Queues**: ADT Stack and its operations: Algorithms and their complexity analysis, Applications of Stacks: Expression Conversion and evaluation– corresponding algorithms and complexity analysis. ADT queue, Types of Queue: Simple Queue, Circular Queue, Priority Queue; Operations on each types of Queues: Algorithms and their analysis.

**Linked Lists:** Singly linked lists: Representation in memory, Algorithms of several operations: Traversing, Searching, Insertion into, Deletion from linked list; Linked representation of Stack and Queue, Header nodes, Doubly linked list: operations on it and algorithmic analysis; Circular Linked Lists: all operations their algorithms and the complexity analysis.

**Module 3**: **[12L]**

**Trees:** Basic Tree Terminologies, Different types of Trees: Binary Tree, Threaded Binary Tree, Binary Search Tree, AVL Tree; Tree operations on each of the trees and their algorithms with complexity analysis. Applications of Binary Trees. B Tree, B+ Tree: definitions, algorithms and analysis.

**Graph:** Basic Terminologies and Representations, Graph search and traversal algorithms and complexity analysis.

**Module 4: [8L]**

**Sorting, Searching and Hashing:** Objective and properties of different sorting

algorithms: Selection Sort, Bubble Sort, Insertion Sort, Quick Sort, Merge Sort, Heap Sort; Performance and Comparison among all the methods, Searching: Linear Search and Binary Search Techniques and their complexity analysis. Hashing: Hashing functions, collision resolution techniques.

**Text books:**

1. "Fundamentals of Data Structures of C" by Ellis Horowitz, Sartaj Sahni, Susan Anderson-freed, Universities Press
2. "Data Structures in C" by Aaron M. Tenenbaum, Yedidyah Langsam, Moshe J. Augenstein , Pearson
3. **"**Data Structures with C" by S. Lipschutz, McGraw Hill India

**Reference books:**

1. Algorithms, Data Structures, and Problem Solving with C++", Illustrated Edition by Mark Allen Weiss, Addison-Wesley Publishing Company
2. "How to Solve it by Computer", 2nd Impression by R.G. Dromey, Pearson Education.

**Course outcomes:**

On completion of this course, students will be able to

1. Implement, analyze and determine the time and space complexity for a given problem of Array, Stack, Queue and Linked list.
2. Implement Tree and Graph and use them in solving a problem.
3. Write and implement an algorithm for Selection Sort, Bubble Sort, Insertion Sort, Quick Sort, Merge Sort, Heap Sort, Linear Search, Binary Search and compare their performance in term of Space and Time complexity.
4. Identify appropriate data structure & algorithmic methods in problem solving.

| **YEAR- 2ND** | | **SEMESTER- III** | |
|---|---|---|---|
| **ES-CS 301** | **Digital Electronics** | **3L:0T: 0P** | **3** |
| **Pre-requisites** | | | |

**Course Objectives**

The objectives are to study

1. To acquire the basic knowledge of digital logic levels and application of knowledge to understand digital electronics circuits.
2. To prepare students to perform the analysis and design of various digital electronic circuits.

**Module 1 :**Fundamentals of Digital Systems and logic families (7 Hours) Digital signals, digital circuits, AND, OR, NOT, NAND, NOR and Exclusive-OR operations, Boolean algebra, examples of IC gates, number systems-binary, signed binary, octal hexadecimal number, binary arithmetic, one's and two's complements arithmetic, codes, error detecting and correcting codes, characteristics of digital lCs, digital logic families, TTL, Schottky TTL and CMOS logic, interfacing CMOS and TTL, Tri-state logic. 8hr

**Module 2:** Combinational Digital Circuits (7 Hours) Standard representation for logic functions, K-map representation, simplification of logic functions using K-map, minimization of logical functions. Don't care conditions, Multiplexer, De-Multiplexer/Decoders, Adders, Subtractors, BCD arithmetic, carry look ahead adder, serial adder, ALU, elementary ALU design, popular MSI chips, digital comparator, parity checker/generator, code converters, priority encoders, decoders/drivers for display devices, Q-M method of function realization. 7hr

**Module 3:** Sequential circuits and systems (7 Hours) A 1-bit memory, the circuit properties of Bi-stable latch, the clocked SR flip flop, J- K-T and D – types flip-flops, applications of flip-flops, shift registers, applications of shift registers, serial to parallel converter, parallel to serial converter, ring counter, sequence generator, ripple(Asynchronous) counters, synchronous counters, counters design using flip flops, special counter IC's, asynchronous sequential counters, applications of counters. 7hr

**Module 4:** A/D and D/A Converters (7 Hours) Digital to analog converters: weighted resistor/converter, R-2R Ladder D/A converter, specifications for D/A converters, examples of D/A converter lCs, sample and hold circuit, analog to digital converters: quantization and encoding, parallel comparator A/D converter, successive approximation A/D converter, counting A/D converter, dual slope A/ D converter, A/D converter using voltage to frequency and voltage to time conversion, specifications of A/D converters, example of A/D converter ICs 7hr

**Module 5:** Semiconductor memories and Programmable logic devices. (7 Hours) Memory organization and operation, expanding memory size, classification and characteristics of memories, sequential memory, read only memory (ROM), read and write memory(RAM), content addressable memory (CAM), charge de coupled device memory (CCD), commonly used memory chips, ROM as a PLD, Programmable logic array, Programmable array logic, complex Programmable logic devices (CPLDS), Field Programmable Gate Array (FPGA). 7hr

Text/References:

1. M. M. Mano, "Digital logic and Computer design", Pearson Education India,

2016.
2. R. P. Jain, "Modern Digital Electronics", McGraw Hill Education, 2009.
3. A. Kumar, "Fundamentals of Digital Circuits", Prentice Hall India, 2016.

**Course Outcomes:**

At the end of this course, students will demonstrate the ability to

- Understand working of logic families and logic gates
- Design and implement Combinational and Sequential logic circuits.
- Understand the process of Analog to Digital conversion and Digital to Analog conversion.
- Be able to use PLDs to implement the given logical problem.

**IT Workshop - (Python/R/MATLAB/Sci Lab)**

| | |
|---|---|
| **Subject Code:** PCC-CS303/PCC-CS 393 | **Category:** Professional core course |
| **Course Title:** IT Workshop - (Python/R/MATLAB/Scilab) | **Semester:** III |
| 1L : 0T: 4P | **Credits:** 3 |
| **Pre-requisites:** No-prerequisite | |

[L= Lecture, T = Tutorials, P = Practicals]

**Programming with Python**

**Objectives of the course:**

This course will expose students to the following

1. Learn core python scripting syntax and semantics
2. Learn fundamental problem solving and programming techniques.
3. Illustrate the need to work with python data structures and their implementation.
4. Learn to work with files, modules, and handle exceptions efficiently.

**Detailed contents:**

| Module | Content | No. of Lecture |
|---|---|---|
| 1 | **Introduction:** History, Features, Setting up path, Working with Python, Basic Syntax, Variable and Data Types, Operator<br>**Conditional Statements**: If, If- else, Nested if-else, Looping, For, While, Nested loops<br>**Control Statements**: Break, Continue, Pass | 3 |
| 2 | **String Manipulation**: Accessing Strings, Basic Operations, String slices, Function and Methods<br>**Functions**: Defining a function, Calling a function, Types of | 3 |

| | | |
|---|---|---|
| | functions, Function Arguments, Anonymous functions, Global and local variables | |
| 3 | **Lists**: Introduction, Accessing list, Operations, Working with lists, Function and Methods<br>**Dictionaries**: Introduction, Accessing values in dictionaries, Working with dictionaries, Properties<br>**Tuple**: Introduction, Accessing tuples, Operations, Working, Functions and Methods | 3 |
| 4 | **Modules**: Importing module, Math module, Random module, Packages, Composition, Input-Output Printing on screen, Reading data from keyboard, Opening and closing file, Reading and writing files.<br>**Exception Handling**: Exception, Exception Handling, Try clause, Except clause. | 3 |

**Suggested text books:**

1. "Introducing Python: Modern Computing in Simple Packages"- O'Reilly Media by Bill Lubanovic.

**Suggested reference books:**

1. "Python Cookbook: Recipes for Mastering Python 3" by David Beazley, Brian K. Jones.
2. "Think Python: How to Think Like a Computer Scientist" by Allen B. Downey.

**Course outcomes:**

1. Write, Test and Debug Python Programs, interpret correct syntax and semantics, and implement conditional branching, iteration.
2. Implement and handle string efficiently, decompose a problem into functions and implement a program using function efficiently.
3. Implement and manipulate a python program using python data structure list, tuple and dictionaries.
4. Implement a python program to read and write data from & to files, handle exceptions, and import modules.

**<u>Programming in R</u>**

**Objectives of the course:**

This course will expose students to the following

1. Learn basic programming syntax and semantics.
2. Learn fundamental problem solving and programming techniques.

**Detailed contents:**

| Module | Content | No. of Lecture |
|---|---|---|
| 1 | Introduction to mechanism for statistics, data analysis, and machine learning; Introduction of R Programming, How to install and run R, Use of R help files, R Sessions, R Objects – Vectors, Attributes, Matrices, Array, Class, List, Data Frames etc. Operators in R. | 3 |
| 2 | R Programming Structures, Control Statements, Loops, Repeat and Break, R-Function, R-Vector Function, Recursive Function in R. | 3 |
| 3 | R Packages (Install and Use), Input/Output Features in R, Reading or Writing in File. Data Manipulation in R. Rearranging data , Random Number and Simulation, Statistical methods like min, max, median, mean, length, Linear Regression, Normal Distribution, Decision tree. | 3 |
| 4 | Graphics, Creating Graphs, The Workhorse of R Base Graphics, Graphical Functions – Customizing Graphs, Saving Graphs to Files, Pie chart, Bar Chart, Histogram. | 3 |

**Suggested Text Book:**

1. "Hands-On Programming with R: Write Your Own Functions and Simulations" by Garrett Grolemund.

**Course outcomes:**

5. Write, Test and Debug R Programs, interpret correct syntax and semantics, and R objects.
6. Implement control flow statements, loop and function.
7. Write program using packages and statistical methods.
8. Understand R graphics.

**Programming in Matlab**

**Objectives of the course:**

This course will expose students to the following

1. Learn the core syntax and semantics
2. Learn fundamental problem solving and programming techniques.

**Detailed contents:**

| Module | Content | No. of Lecture |
|---|---|---|
| 1 | **Introduction:** Why MATLAB? , History, Its strengths, Competitors, Starting MATLAB, Using MATLAB as a calculator, Quitting MATLAB<br>**Basics**: Familiar with MATLAB windows, Basic Operations, MATLAB-Data types, Rules about variable names, Predefined variables | 3 |
| 2 | **Programming-I:** Vector, Matrix, Array Addressing, Built-in functions, Mathematical Operations, Dealing with strings (Array of characters), Array of array (cell) concept<br>**Programming-II**: Script file, Input commands, Output commands, Structure of function file, Inline functions, Feval command, Comparison between script file and function file | 3 |
| 3 | **Conditional statements and Loop:** Relational and Logical Operators , If-else statements, Switch-case statements, For loop, While loop, Special commands (Break and continue), Import data from large database, Export data to own file or database | 3 |
| 4 | **2D Plotting**: In-built functions for plotting, Multiple plotting with special graphics, Curve fitting, Interpolation, Basic fitting interface<br>**3D Plotting:** Use of meshgrid function, Mesh plot, Surface plot, Plots with special graphics | 3 |

**Suggested Text Book:**

1. "Matlab: A Practical Introduction to Programming and Problem Solving" by Stormy Attaway.

**Course outcomes:**

1. Write, Test and Debug Matlab Programs, interpret correct syntax and semantics.
2. Implement conditional branching, loop, and iteration.
3. Implement vector, matrix, array, function and file.
4. Apply 2D and 3D plotting.

**<u>Programming in Scilab</u>**

**Objectives of the course:**

This course will expose students to the following

1. Learn the core syntax and semantics
2. Learn fundamental problem solving and programming techniques.

**Detailed contents:**

| Module | Content | No. of Lecture |
|---|---|---|
| 1 | **Introduction:** About Scilab, History, Its strengths, Competitors, Starting Scilab, Using Scilab as a calculator, Quitting Scilab<br>**Basics**: Scilab datatypes, variables and constants. | 3 |
| 2 | **Matrices and Arrays:** Matrix and basic matrix operations of addition and multiplication, transpose, Determinants of 2 × 2 and 3×3 matrices, and array.<br>**Expression:** Variables Numbers, Operators Functions, Expressions. | 3 |
| 3 | **Conditional statements and Loop:** If, else, else if, switch, for, while, continue, break, try, catch, return, etc. | 3 |
| 4 | **Graphics:** Plotting Process, Editing Process, Preparing Graphs, Basic Plotting Functions, Plotting math functions in Scilab, Plotting Bar graphs in Scilab, Changing axes properties in scilab plots, Image Reading & Writing, Printing graphics. | 3 |

**Suggested Text Book:**

1."Scilab from Theory to Practice - I. Fundamentals" by Philippe Roux.

**Course outcomes:**

1. Write, Test and Debug Scilab Programs, interpret correct syntax and semantics.
2. Implement conditional branching, loop, and iteration.
3. Implement matrix and array.
4. Apply plotting function and graphics.

| | |
|---|---|
| Course Code: **BS-M301** | Category: Basic Science Courses |
| **Course Title: Mathematics-III (Differential Calculus)** | Semester: 3rd Semester |
| L-T-P: 2-0-0 | Credit: 2 |
| **Teaching Scheme**<br>Theory:2 hrs./week<br>Tutorial: NIL<br>Practical: NIL | **Examination Scheme**<br>Class Tests & Assignments: 25 marks<br>Attendance: 5 marks<br>End Semester Exam: 70 marks |
| **Objective:**<br>Objective:<br>1 To know Convergence of sequence and series<br>2 To know Limit, continuity and partial derivatives, Chain rule, Implicit function<br>3 To know First Order Differential Equation, Exact, Linear and Bernoulli'sequations, Basic Concept of graph, Walk, Path Circuit, Euler and Hamiltonian graph, diagraph. | |
| **Pre-requisites:**<br>1 Concept Linear Algebra Determinant and its properties (up to third order)<br>2 Minor and cofactors, Matrices, addition, multiplication and transpose of a matrix,Symmetric and skew-symmetric | |

| Unit | Content | Hrs/Unit |
|---|---|---|
| 1 | Convergenceandsequence series, tests forconvergence, power series, Taylor's series. Series for exponential trigonometricand logarithmicfunctions. | 8 |
| 2 | Limit, continuity and partial derivatives, Chain rule,Implicit function,Jacobian, Directional derivatives,Total derivative; Maxima, minima and saddle points;Gradient, curl and divergence and related problems. | 7 |
| 3 | Double and triple integrals (Cartesian and polar),change of order of integration in double integrals,Change of variables (Cartesian to polar). Theoremsof Green, Gauss and Stokes (Statement only) andrelated problems. | 8 |
| 4. | First Order Differential Equation, Exact, Linear andBernoulli's equations, Equations of first order butnot of first degree: equations solvable for p,equations solvable for y, equations solvable for x. Second order linear differential equations with constant coefficients, D-operator method, method of variation of | 9 |

| | | |
|---|---|---|
| | parameters, Cauchy-Euler equation. [4L] | |
| 5 | Basic Concept of graph, Walk, Path Circuit, Euler and Hamiltonian graph, diagraph.<br>Matrix Representation: Incidence & Adjacency matrix. Tree: Basic Concept of tree, Binary tree, Spanning Tree, KrusKal and Prim's algorithm for finding the minimal spanning tree. | 8 |

Text book and Reference books:


1. Higher Algebra, S. K. Mapa, Levant Books.
2. Advanced Higher Algebra, Chakravorty and Ghosh, U N Dhar Pvt. Ltd.
3. Co-ordinate Geometry, S. L. Loney
4. Integral Calculus, Das and Mukherjee, U N Dhar Pvt. Ltd.
5. Differential Calculus, Das and Mukherjee, U N Dhar Pvt. Ltd.
6. Advanced Engineering Mathematics, E Kreyszig
7. Advanced Engineering Mathematics, Chandrika Prasad & Reena Garg, Khanna Publishing House (AICTE Recommended Textbook -2018)


**On completion of the course students will be able to**

BSC-301.1 Express a logic sentence in terms of predicates, quantifiers, and logical connectives.

BSC-301.2 Apply the rules of inference and methods of proof including direct and indirect proof forms, proof by contradiction, and mathematical induction.

BSC-301.3 Use tree and graph algorithms to solve problems BSC-301.4 Evaluate Boolean functions and simplify expressions using the properties of boolean algebra.

| YEAR- 2ND | | SEMESTER- III | |
|---|---|---|---|
| PCC-CS 391 | Principles of Programming Language Lab | 0L:0T:4P | Credit 2 |
| Pre-requisites | | | |

## Course Code: PCC-CS 391
## Course Name: Principles of Programming Languages Lab (PPL Lab)
## Credits: 2 (Lab: 4hrs/week)

**Course Objectives:**

This is the Practical part associated with the PPL course (PCC-CS 301). The objective is to impart a hands on exposure to the key concepts introduced in PCC-CS 301.

**Course Outcomes:**

After successful completion of this course, students should be able to

1. Write simple programs using an OO programming language such as C++/Python/Ruby
2. Write simple programs using a logic programming language such as Prolog/Lisp
3. Write simple programs using a functional programming language such Haskell/ML/SML/Scheme/Clojure

**Syllabus:**

Introduction to OO programming concepts using Python/Ruby/C++ [~4 weeks]

Introduction to Functional programming using SML/Ocaml/Haskell/Scheme [~4 weeks]

Introduction to Logic programming using Prolog/Lisp [~4 weeks]

**Textbook:**

Programming Language Pragmatics by Michael Scott. Fourth Edition, Morgan Kaufmann Publishers, 2015. ISBN-13: 978-0124104099

Relevant online documentation for the respective programming languages used.

| YEAR- 2ND | | SEMESTER- III | |
|---|---|---|---|
| ES-CS 391 | Digital Electronics Laboratory | 0L:0T: 4P | Credit 2 |
| Pre-requisites | | | |

**List of Experiments**

1. Realization of basic gates using Universal logic gates.
2. Code conversion circuits- BCD to Excess-3 and vice-versa.
3. Four-bit parity generator and comparator circuits.
4. Construction of simple Decoder and Multiplexer circuits using logic gates.
5. Design of combinational circuit for BCD to decimal conversion to drive 7-segment display using multiplexer.
6. Construction of simple arithmetic circuits- Adder, Subtractor.
7. Realization of RS-JK and D flip-flops using Universal logic gates.
8. Realization of Universal Register using JK flip-flops and logic gates.
9. Realization of Universal Register using multiplexer and flip-flops.
10. Realization of Asynchronous Up/Down counter.
11. Realization of Synchronous Up/Down counter.
12. Realization of Ring counter and Johnson's counter.
13. Construction of adder circuit using Shift Register and full Adder.
14. Code conversion circuits – Binary to Gray & Vice-Versa. **(Innovative).**
15. Design of Sequential Counter with irregular sequences. **(Innovative)**

| YEAR- 2ND | | SEMESTER- III | |
|---|---|---|---|
| PCC-CS392 | Data Structure & Algorithms LAB | 0L:0T: 4P | Credit: 2 |
| Pre-requisites | ES-CS201 and ES-CS291 | | |

**Detailed contents:**

**Experiments should include but not limited to :**

1. Implementation of array operations; Merging of two arrays; Sparse Matrices : Addition, Multiplication.
2. Stacks and Queues: adding, deleting elements; Circular Queue: Adding & deleting elements; Evaluation of expressions;
3. Implementation of linked lists: inserting, deleting, inverting a linked list; Implementation of stacks & queues using linked lists; Polynomial addition, Polynomial multiplication
4. Recursive and Nonrecursive traversal of binary trees; Threaded binary tree traversal; AVL tree implementation; Application of Trees.
5. Graph implantation and traversal.
6. Searching & sorting techniques; Application of sorting and searching algorithms
7. Hash tables implementation: searching, inserting and deleting,

**COMPUTER SCIENCE AND ENGINEERING**

## Semester-IV (2nd Year)

| | |
|---|---|
| Course Code: PCC-CS401 | Category: Professional Core Courses |
| **Course Title: Discrete Mathematics** | Semester: 4th Semester |
| L-T-P: 3-1-0 | Credit: 4 |
| **Teaching Scheme**<br>Theory: 3hrs/week<br>Tutorial: 1hr/week | **Examination Scheme**<br>Class Tests & Assignments: 25 marks<br>Attendance: 5 marks<br>End Semester Exam: 70 marks |
| **Objective:**<br>Throughout this course, students will be expected to demonstrate their understanding of Discrete Mathematics by being able to do each of the following:<br>1. Use mathematically correct terminology and notation.<br>2. Construct correct direct and indirect proofs.<br>3. Use division into cases in a proof.<br>4. Use counterexamples.<br>5. Apply logical reasoning to solve a variety of problems. | |
| **Pre-requisites:** Some concepts from basic math – algebra, geometry, pre-calculus | |

***Detailed Syllabus***

| Unit | Content | Hrs/Unit (L+T) | Marks/Unit |
|---|---|---|---|
| 1 | **Sets, Relation and Function:** Operations and Laws of Sets, Cartesian Products, Binary Relation, Partial Ordering Relation, Equivalence Relation, Image of a Set, Sum and Product of Functions, Bijective functions, Inverse and Composite function, Size of a Set, Finite and infinite Sets, Countable and uncountable Sets, Cantor's diagonal argument and The Power Set theorem, Schroeder-Bernstein theorem.<br><br>**Principles of Mathematical Induction:** The Well-Ordering Principle, Recursive definition, The Division algorithm: Prime Numbers, The Greatest Common Divisor: Euclidean Algorithm, The Fundamental Theorem of Arithmetic. | 8+3 | |
| 2 | **Basic Counting Techniques:** Inclusion and Exclusion, Pigeon-Hole Principle, Permutation and Combination. | 5+1 | |

| | | | |
|---|---|---|---|
| 3 | **Propositional Logic:** Syntax, Semantics, Validity and Satisfiability, Basic Connectives and Truth Tables, Logical Equivalence: The Laws of Logic, Logical Implication, Rules of Inference, The use of Quantifiers. **Proof Techniques:** Some Terminology, Proof Methods and Strategies, Forward Proof, Proof by Contradiction, Proof by Contraposition, Proof of Necessity and Sufficiency. | 8+3 | |
| 4 | **Algebraic Structures and Morphism:** Algebraic Structures with one Binary Operation, Semi Groups, Monoids, Groups, Congruence Relation and Quotient Structures, Free and Cyclic Monoids and Groups, Permutation Groups, Substructures, Normal Subgroups, Algebraic Structures with two Binary Operation, Rings, Integral Domain and Fields. Boolean Algebra and Boolean Ring, Identities of Boolean Algebra, Duality, Representation of Boolean Function, Disjunctive and Conjunctive Normal Form. | 7+2 | |
| 5 | **Graphs and Trees:** Graphs and their properties, Degree, Connectivity, Path, Cycle, Sub Graph, Isomorphism, Eulerian and Hamiltonian Walks, Graph Colouring, Colouring maps and Planar Graphs, Colouring Vertices, Colouring Edges, List Colouring, Perfect Graph, definition properties and Example, Rooted Trees, Trees and Sorting, Weighted Trees and Prefix Codes, Bi-connected component and Articulation Points, Shortest distances. | 8+3 | |

***Suggested Text books***


1. Kenneth H. Rosen, "Discrete Mathematics and its Applications", Tata McGraw Hill
2. Susanna S. Epp, "Discrete Mathematics with Applications", 4th edition, Wadsworth Publishing Co. Inc.
3. C L Liu and D P Mohapatra, "Elements of Discrete Mathematics A Computer Oriented Approach", 3rd Edition, Tata McGraw Hill.


***Suggested Reference books***


1. J.P. Tremblay and R. Manohar, "Discrete Mathematical Structure and It's Application to Computer Science", Tata McGraw Hill

2. Norman L. Biggs, Discrete Mathematics, 2nd Edition, Oxford University Press
3. Seymour Lipschutz, Marc Lipson, "Schaum's Outline of Theory and Problems of Discrete Mathematics", Tata McGraw Hill

***Course Outcomes***

On completion of the course students will be able to:

PCC-CS401.1 For a given logic sentence express it in terms of predicates, quantifiers, and logical connectives

PCC-CS401.2 For a given a problem, derive the solution using deductive logic and prove the solution based on logical inference

PCC-CS401.3 For a given a mathematical problem, classify its algebraic structure

PCC-CS401.4 Evaluate Boolean functions and simplify expressions using the properties of Boolean algebra

PCC-CS401.5 Develop the given problem as graph networks and solve with techniques of graph theory.

**Computer Organization & Architecture**

**Subject Code:** PCC-CS402

| | |
|---|---|
| **Subject Code:** PCC-CS402 | **Category:** Professional core course |
| **Course Title:** Computer Organization & Architecture | **Semester:** IV |
| 3L : 0T: 0P | **Credits:** 3 |
| **Pre-requisites:** ES-CS301 | |

[L= Lecture, T = Tutorials, P = Practicals]

**Objectives of the course:**

This course will expose students to the following

1. How Computer Systems work & the basic principles.
2. Instruction Level Architecture and Instruction Execution.
3. The state of art in memory system design.
4. How I/O devices are accessed and its principles.
5. To provide the knowledge on Instruction Level Parallelism.
6. To impart the knowledge on micro programming.
7. Concepts of pipelining techniques.

**Detailed contents:**

| Module | Content | No. of Lecture |
|---|---|---|
| 1 | **Functional blocks of a computer:** CPU, memory, input-output | 10 |

| | | |
|---|---|---|
| | subsystems, control unit. Instruction set architecture of CPU– registers, instruction execution cycle, addressing modes, instruction set.<br><br>**Data representation:** Signed number representation, fixed and floating point representations, character representation. Computer arithmetic –integer addition and subtraction, ripple carry adder, carry look-ahead adder, etc. multiplication – shift-and add, Booth multiplier, Division restoring and non-restoring techniques, Floating point - IEEE 754 standard, floating point arithmetic. | |
| 2 | **Memory organization:** Memory interleaving, concept of hierarchical memory organization, semiconductor memory technologies, cache memory, cache size vs. block size, mapping functions, replacement algorithms, write policies. | 10 |
| 3 | Introduction to x86 architecture.<br>**CPU control unit design:** hardwired and micro-programmed design approaches.<br>**Peripheral devices and their characteristics:** Input-output subsystems, I/O device interface, I/O transfers–program controlled, interrupt driven and DMA, interrupts. | 8 |
| 4 | **Pipelining:** Basic concepts of pipelining, throughput and speedup, pipeline hazards.<br>**Parallel Processors:** Introduction to parallel processors. | 8 |

**Suggested text books:**

1. "Computer Organization and Design: The Hardware/Software Interface", 5th Edition by David A. Patterson and John L. Hennessy, Elsevier.
2. "Computer Organization and Embedded Systems", 6th Edition by Carl Hamacher, McGraw Hill Higher Education.
3. "Computer System Architecture", by M. Morris Mano.
4. " Computer Architecture", Oxford University Press by Behrooz Parhami

**Suggested reference books:**

1. "Computer Architecture and Organization", 3rd Edition by John P. Hayes, WCB/McGraw-Hill.
2. "Computer Organization and Architecture: Designing for Performance", 10th Edition by William Stallings, Pearson Education.
3. "Computer System Design and Architecture", 2nd Edition by Vincent P. Heuring and Harry F. Jordan, Pearson Education.

**Course outcomes:**

1. Draw the functional block diagram of single bus architecture of a computer and describe the function of the instruction execution cycle, addressing modes, instruction set.
2. Given a CPU organization and instruction, design a memory module; analyze its operation by interfacing with the CPU and mapping techniques.
3. Write assembly language program for specified microprocessor, ability to explain the concept of control unit and I/O operations.
4. Given a CPU organization, apply design techniques to enhance performance using pipelining, and understand the concept of parallelism.

**OPERATING SYSTEMS**
**PCC-CS403**
**3L:0T:0P 3 Credits**
**Total Number of Lectures: 36**

| | |
|---|---|
| **Course Code** | PCC-CS403 |
| **Course Name** | OPERATING SYSTEMS |
| **Credits** | 3L:0T:0P 3 Credits |
| **Pre-Requisites** | Basic understanding of Computers, Data structure & Algorithms |

**Total Number of Lectures: 36**

| **COURSE OBJECTIVES** |
|---|
| **1.** To learn the mechanisms of OS to handle processes and threads and their communication |
| **2.** To learn the mechanisms involved in memory management and virtual memory in contemporary OS |
| **3.** To gain knowledge on deadlock related issues |
| **4.** To know the concepts of I/O, File and Disk Management |

**Syllabus:**

**Module 1:**

**Introduction:** Concept of Operating Systems, Generations of Operating systems, Types of Operating Systems, OS Services, System Calls, Structure of an OS - Layered, Monolithic, Microkernel Operating Systems, Concept of Virtual Machine. Case study on UNIX and WINDOWS Operating System.

**Module 2:**

**Processes:** Definition, Process Relationship, Different states of a Process, Process State transitions, Process Control Block (PCB), Context switching

**Thread:** Definition, Various states, Benefits of threads, Types of threads, Concept of multithreads,

**Process Scheduling:** Foundation and Scheduling objectives, Types of Schedulers, Scheduling criteria: CPU utilization, Throughput, Turnaround Time, Waiting Time, Response Time; Scheduling algorithms: Pre-emptive and Non pre-emptive, FCFS, SJF, RR; Multiprocessor scheduling: Real Time scheduling: RM and EDF.

**Module 3:**

**Inter-process Communication:** Critical Section, Race Conditions, Mutual Exclusion, Hardware Solution, Strict Alternation, Peterson's Solution, The Producer-Consumer Problem, Semaphores, Event Counters, Monitors, Message Passing, Classical IPC Problems: Reader's & Writer Problem, Dinning Philosopher Problem etc.

**Module 4:**

**Memory Management:** Basic concept, Logical and Physical address map, Memory allocation: Contiguous Memory allocation – Fixed and variable partition– Internal and External fragmentation and Compaction; Paging: Principle of operation – Page allocation – Hardware support for paging, Protection and sharing, Disadvantages of paging.

**Module 5:**

**Virtual Memory:** Basics of Virtual Memory – Hardware and control structures – Locality of reference, Page fault , Working Set , Dirty page/Dirty bit – Demand paging, Page Replacement algorithms: Optimal, First in First Out (FIFO), Second Chance (SC), Not recently used (NRU) and Least Recently used (LRU).

**Module 6:**

**Deadlocks:** Definition, Necessary and sufficient conditions for Deadlock, Deadlock Prevention, Deadlock Avoidance: Banker's algorithm, Deadlock detection and Recovery.

**Module 7:**

**I/O Management:** I/O hardware, polling, interrupts, DMA, application I/O interface (block and character devices, network devices, clocks and timers, blocking and nonblocking I/O), kernel I/O subsystem (scheduling, buffering, caching, spooling and device reservation, error handling), performance.

**File Management:** Concept of File, Access methods, File types, File operation, Directory structure, File System structure, Allocation methods (contiguous, linked, indexed), Free-space management(bit vector, linked list, grouping), directory implementation (linearlist, hash table),efficiency and performance.

**Disk Management:** Disk structure, Disk scheduling - FCFS, SSTF, SCAN, C-SCAN, Disk reliability, Disk formatting, Boot-block, Bad blocks

| **LECTURE WITH BREAKUP** | **NO. OF LECTURES** |
|---|---|
| **Module 1:**<br><br>**Introduction:** Concept of Operating Systems, Generations of Operating systems, Types of Operating Systems, OS Services, System Calls, Structure of an OS - Layered, Monolithic, Microkernel Operating Systems, Concept of Virtual Machine. Case study on UNIX and WINDOWS Operating System. | **[3]** |
| **Module 2:**<br><br>**Processes:** Definition, Process Relationship, Different states of a Process, Process State transitions, Process Control Block (PCB), Context switching<br><br>**Thread:** Definition, Various states, Benefits of threads, Types of threads, Concept of multithreads,<br><br>**Process Scheduling:** Foundation and Scheduling objectives, Types of Schedulers, Scheduling criteria: CPU utilization, Throughput, Turnaround Time, Waiting Time, Response Time; Scheduling algorithms: Pre-emptive and Non pre-emptive, FCFS, SJF, RR; Multiprocessor scheduling: Real Time scheduling: RM and EDF. | **[4]** |

| | |
|---|---|
| **Module 3:**<br><br>**Inter-process Communication:** Critical Section, Race Conditions, Mutual Exclusion, Hardware Solution, Strict Alternation, Peterson's Solution, The Producer-Consumer Problem, Semaphores, Event Counters, Monitors, Message Passing, Classical IPC Problems: Reader's & Writer Problem, Dinning Philosopher Problem etc. | **[4]** |
| **Module 4:**<br><br>**Memory Management:** Basic concept, Logical and Physical address map, Memory allocation: Contiguous Memory allocation – Fixed and variable partition– Internal and External fragmentation and Compaction; Paging: Principle of operation – Page allocation – Hardware support for paging, Protection and sharing, Disadvantages of paging. | **[6]** |
| **Module 5:**<br><br>**Virtual Memory:** Basics of Virtual Memory – Hardware and control structures – Locality of reference, Page fault , Working Set , Dirty page/Dirty bit – Demand paging, Page Replacement algorithms: Optimal, First in First Out (FIFO), Second Chance (SC), Not recently used (NRU) and Least Recently used (LRU). | **[6]** |
| **Module 6:**<br><br>**Deadlocks:** Definition, Necessary and sufficient conditions for Deadlock, Deadlock Prevention, Deadlock Avoidance: Banker's algorithm, Deadlock detection and Recovery. | **[6]** |
| **Module 7:**<br><br>**I/O Management:** I/O hardware, polling, interrupts, DMA, application I/O interface (block and character devices, network devices, clocks and timers, blocking and nonblocking I/O), kernel I/O subsystem (scheduling, buffering, caching, spooling and device reservation, error handling), performance.<br><br>**File Management:** Concept of File, Access methods, File types, File operation, Directory structure, File System structure, Allocation methods (contiguous, linked, indexed), Free-space management (bit vector, linked list, grouping), directory implementation (linear list, hash table), efficiency and performance.<br><br>**Disk Management:** Disk structure, Disk scheduling - FCFS, SSTF, | **[7]** |

| | |
|---|---|
| SCAN, C-SCAN, Disk reliability, Disk formatting, Boot-block, Bad blocks | |

| **COURSE OUTCOMES** |
|---|
| **After completion of course, students would be able to:** |
| CO1: Create processes and threads and analyse the concepts of processes and threads in operating system and illustrate the scheduling of processor for a given problem instance. |
| CO2: For a given specification of memory organization, develop the techniques for optimally allocating memory to processes by increasing memory utilization and for improving the access time. |
| CO3: Identify and handle deadlock related issues. |
| CO4: Understand the implement file systems and directories along with the interfacing of IO devices with the operating system and disk management. |

**Textbooks/References:**

**Suggested books:**


1. Operating System Concepts Essentials, 9th Edition by Avi Silberschatz, Peter Galvin, Greg Gagne, Wiley Asia Student Edition.
2. Operating Systems: Internals and Design Principles, 5th Edition, William Stallings, Prentice Hall of India.


**Suggested reference books:**


1. Operating System: A Design-oriented Approach, 1st Edition by Charles Crowley, Irwin Publishing
2. Operating Systems: A Modern Perspective, 2nd Edition by Gary J. Nutt, Addison-Wesley
3. Design of the Unix Operating Systems, 8th Edition by Maurice Bach, Prentice-Hall of India
4. Understanding the Linux Kernel, 3rd Edition, Daniel P. Bovet, Marco Cesati, O'Reilly and Associates

**Design and Analysis of Algorithms**
**Course Code: PCC-CS404**

| | |
|---|---|
| **Course Code** | **PCC-CS404** |
| **Course Name** | **Design and Analysis of Algorithms 3L:0T: 0P** |
| **Credits** | 3 |
| **Pre-Requisites** | ES-CS201 and PCC-CS302 |

Total Number of Lectures:36

| **COURSE OBJECTIVE** |
|---|
| • Analyze the asymptotic performance of algorithms. |
| • Demonstrate a familiarity with major algorithm paradigms, their analysis and applications to different problem domains. |
| • To classify problems according to hardness and to solve computationally hard problems. |

| Lecture Distributions | Hours |
|---|---|
| **Unit 1**<br>Introduction: Characteristics of algorithm: Correctness and Efficiency: Case Study<br>Asymptotic analysis, Notations and their properties: Complexity bounds – Upper, lower and tight bound: Case Study<br>Design Strategy: Sorting by Divide and Conquer: Heap Sort<br>Performance measurements of Algorithm: Time complexity analysis: Sorting: Lower bound: Matrix Multiplications<br>Analysis of recursive algorithms through recurrence relations: Substitution method: Case study: Recursion tree method and Masters' theorem: Case study | 10 |
| **Unit 2**<br>Fundamental Algorithmic Strategies: Brute-Force: Greedy: Dynamic Programming: Branch-and-Bound and Backtracking methodologies for the design of algorithms: Illustrations of these techniques for Problem-Solving: Bin Packing: Knap Sack:TSP: Heuristics –characteristics and their application domains | 9 |
| **Unit 3**<br>Graph and Tree Algorithms: Traversal algorithms: Depth First Search (DFS) and Breadth | 9 |

| | |
|---|---|
| First Search (BFS): Shortest path algorithms: Transitive closure: Minimum Spanning Tree:<br>Topological sorting: Network Flow Algorithms | |
| **Unit 4**<br>Tractable and Intractable Problems: Computability classes – P, NP, NP-complete and NP-hard: Cook's theorem: Standard NP-complete problems and Reduction techniques | 4 |
| **Unit 5**<br>Advanced Topics: Approximation algorithms: Vertex cover: Set Cover:<br>TSP: Randomized algorithms: Class of problems beyond NP – P SPACE | 4 |

| **COURSE OUTCOMES** |
|---|
| After completion of course, students would be able to: |
| 1. Analyze the worst-case running times of algorithms based on asymptotic analysis. |
| 2. Model a problem and develop the appropriate algorithm from divide and conquer, greedy, dynamic programming and other paradigms |
| 3. Classify problems in appropriate complexity classes and apply approximation and randomized algorithms in solving computationally hard real life problems |

Suggested books:

1. Introduction to Algorithms, 4TH Edition, Thomas H Cormen, Charles E Lieserson, Ronald L Rivest and Clifford Stein, MIT Press/McGraw-Hill.
2. Fundamentals of Algorithms – E. Horowitz et al.

**Suggested reference books**

1. Algorithm Design, 1ST Edition, Jon Kleinberg and ÉvaTardos, Pearson.
2. Algorithm Design: Foundations, Analysis, and Internet Examples, Second Edition,
Michael T Goodrich and Roberto Tamassia, Wiley.
3. Algorithms -- A Creative Approach, 3RD Edition, UdiManber, Addison-Wesley,Reading, MA.

**Organizational Behaviour**

**Course Code: HSMC-401**

| | |
|---|---|
| **Course Code** | **HSMC-401** |
| **Course Name** | **Management-I (Organizational Behaviour/ Finance & Accounting)** |
| **Credits** | 3 |
| **Pre-Requisites** | |

Objective: the objective of this course is to identify and understand the importance and significance of human behaviour in the organizational setting. It also illuminates the philosophy, tools and relevance of organizational behaviour and its role for individual and organizational growth.

| Sl. | Contents | **Lecture Hour** |
|---|---|---|
| 1. | OB – Overview – Meaning of OB, Importance of OB, Field of OB, Contributing Disciplines, Applications in Industry, Role of Managers in OB- Interpersonal Roles-Informational Roles- Decisional Roles, Challenges and Opportunities for OB | 04 |
| 2. | Personality– Meaning of Personality, Determinants of Personality, Theories of Personality, Measurement of Personality, Development of Personality | 04 |
| 3. | Perception – Process and Principles, Nature and Importance, Factors Influencing, Perception, Perceptual Selectivity, Social Perception, Fundamentals of Decision making | 04 |
| | Communication- Meaning and Importance, Process, Function & Types of Effective Communication, Interpersonal Communication, Organizational Communication, | 04 |
| 4. | Motivation in OB – Approaches to Work Motivation, Theories of Motivation – Maslow's Hierarchy of Need Theory, Alderfer's ERG Theory, Herzberg's Motivation-Hygiene Theory, McClelland's Achievement – Motivation Theory, McGregor's Theory X & Y, Vroom's Expectancy Theory, Porter Lawler Expectancy Model | 06 |
| 5. | Attitudes and Job Satisfaction – Sources of Attitudes, Types of Attitudes, Attitudes and Consistency, Cognitive Dissonance Theory, Attitude Surveys. | 02 |
| 6. | Organization - Mission, Goals, Characteristics, Types, Organizational Theory- Classical Theories: Scientific Management, Administrative Principals, Bureaucracy, Human | 06 |

| | | |
|---|---|---|
| | Relation Approach, Modern Theories: System Approach, Contingency Approach, Quantitative Approach, Behavioural Approach, Managing Organizational Culture. | |
| 7. | Group Behaviour - Characteristics of Group, Types of Groups, Stages of Development, Group Decision-making, difference work group and work team, Why work Teams, Work Team in Organization, Team Building, Group Dynamics, Organizational Politics. | 04 |
| 8. | Leadership - Leadership Theories, Leadership Styles, Skills and influence process, Leadership and power, Examples of Effective Organizational Leadership in India, Cases on Leadership, Success stories of today's Global and Indian leaders. | 04 |
| 9. | Conflict in Organization - Sources of Conflict, Types of Conflict, Conflict Process, Johari Window, Conflict Resolution, Cases on Conflict Resolution. | 04 |
| 10. | Organizational Change - Meaning and Nature of Organizational Change, Types of Organizational Change, Forces that act as a stimulant to change. Resistance to change, How to overcome resistance to change, Approaches to Managing Organizational Change, Kurt Lewin's three Step model, Action research model, Kotter's Eight Step model. | 06 |
| | Total | 48 |

**Suggested Reading**


1. Robbins, S.P. Judge, T.A. & Sanghi, S.: Organizational Behaviour, Pearson
2. Luthans, Fred: Organizational Behaviour, McGraw Hill
3. Newstrom J.W. &Devis K.: Organizational Behaviour, McGraw Hill
4. Aswathappa ,K : Organisational Behaviour ,Himalaya Publishing House
5. Shukla, Madhukar : Understanding Organizations – Organizational Theory & Practice in India, Prentice Hall
6. Sekharan, Uma: Organisational Behaviour , The Mc Graw –Hill Companies

**Computer Organization & Architecture Lab**
**Subject Code:** PCC-CS492

| | |
|---|---|
| **Subject Code:** PCC-CS492 | **Category:** Professional core course |
| **Course Title:** Computer Organization & Architecture Lab | **Semester:** IV |
| 0**L** : 0**T**: 4**P** | **Credits:** 2 |

[L= Lecture, T = Tutorials, P = Practicals]

**Objectives of the course:**

This course will expose students to the following

1. To be able to understand the behavior logic gates, adder, decoder and multiplexer.

2. How Computer Systems work & the basic principles.

**Detailed contents:** All laboratory assignments are based on Hardware Description Language (VHDL or Verilog) Simulation.

| **Module** | **Content** |
|---|---|
| 1 | HDL introduction. |
| 2 | Basic digital logic based programming with HDL. |
| 3 | 8-bit Addition, Multiplication, Division. |
| 4 | Design a BCD adder. |
| 5 | Design an 8×1 multiplexer. |
| 6 | Design a Decoder. |
| 7 | 8-bit Register design. |
| 8 | Memory unit design and perform memory operations. |
| 9 | 8-bit simple ALU design. |
| 10 | 8-bit simple CPU design. |
| 11 | Interfacing of CPU and Memory. |

**Course Outcome:**

1. Understand basic logic gates and efficiently verify the behavior of the circuit.
2. Acquire in-depth knowledge to implement different arithmetic operations.
3. Understand the working principle of the memory system.
4. Understand the working principle of CPU, and interfacing of CPU and Memory.

**OPERATING SYSTEMS LAB**
**Course Code: PCC-CS493**

| | |
|---|---|
| **Course Code** | PCC-CS493 |
| **Course Name** | OPERATING SYSTEMS LAB |
| **Credits** | 0L:0T:4P 2 Credits |
| **Pre-Requisites** | Programming for Problem Solving, Data Structure & Algorithm |

**Total Number of Lectures: 48**

| **COURSE OBJECTIVES** |
|---|
| **1.** To write shell scripts, make a script executable, and learn varying shell syntax (variables, conditions, control structures, functions, commands). |
| **2.** To create process and handle the concepts of process, signal and semaphore in operating system. |
| **3.** To understand how to implement thread and thread synchronization, Inter-process communication |

**Syllabus:**

**1.** Shell programming [9P]: creating a script, making a script executable, shell syntax (variables, conditions, control structures, functions, commands).

**2.** Process [6P]: starting new process, replacing a process image, duplicating a process image, waiting for a process, zombie process.

**3.** Signal [9P]: signal handling, sending signals, signal interface, signal sets.

**4.** Semaphore [6P]: programming with semaphores (use functions semctl, semget, semop, set_semvalue, del_semvalue, semaphore_p, semaphore_v).

**5.** POSIX Threads [9P]: programming with pthread functions(viz. pthread_create, pthread_join, pthread_exit, pthread_attr_init, pthread_cancel)

**6.** Inter-process communication [9P]: pipes(use functions pipe, popen, pclose), named pipes(FIFOs, accessing FIFO)

| COURSE OUTCOMES |
|---|
| **After completion of course, students would be able to:** |
| CO1: Use a UNIX text editor to create a shell script and run scripts efficiently from the command line and write varying shell syntax. |
| CO2: Create process and handle the concepts of process, signal and semaphore in operating system. |
| CO3: Implement thread and thread synchronization, Inter-process communication |

**Textbooks/References:**


- Unix Shell Programming, Yahhwant P. Kanetkar
- Unix and Shell Programming, Sumitabha Das
- Advanced Programming in the UNIX Environment 2nd Edition (English, Paperback, W. Richard Stevens)
- Milenkovie M., "Operating System : Concept & Design", McGraw Hill.
- Tanenbaum A.S., "Operating System Design & Implementation", Practice Hall NJ.
- Silbersehatz A. and Peterson J. L., "Operating System Concepts", Wiley.
- Dhamdhere: Operating System TMH
- Stalling, William, "Operating Systems", Maxwell McMillan International Editions, 1992.
- Dietel H. N., "An Introduction to Operating Systems", Addison Wesley


**Design and Analysis of Algorithms Lab**
**Course Code: PCC-CS-494**

| **Course Code** | **PCC-CS-494** |
|---|---|
| **Course Name** | **Design and Analysis of Algorithms Lab** |
| **COURSE OBJECTIVE** | |
| • Implement C-programs in different paradigms of Algorithm Design | |
| • To improve the efficiency of implementation using suitable data structures | |
| • To implement programs for hard problems using approximation algorithms | |

**List of Assignments**
**Lab 1: Sorting Algorithms:**

1. Write a program to implement the Bubble Sort algorithm.
2. Write a program to implement the Merge Sort algorithm.
3. Write a program to implement the Quick Sort algorithm.

**Lab 2: Sorting Algorithms:**

1. Write a program to implement the Insertion Sort algorithm.
2. Write a program to implement the Selection Sort algorithm.
3. Write a program to implement the Heap Sort algorithm.

**Lab 3: Searching Algorithms:**

1. Write a program to implement the Linear Search algorithm.
2. Write a program to implement the Binary Search algorithm.

**Lab 4: Greedy Algorithms:**

1. Write a Program to implement fractional Knapsack problem.

**Lab 5: Minimum Spanning Tree**

1. Write a program to implement PRIMS algorithm to find Minimum cost Spanning Tree (MST).
2. Write a program to implement KRUSKAL algorithm to find Minimum cost Spanning Tree (MST).

**Lab 6: Shortest path algorithms**

1. Write a program to implement Dijkstra's algorithm to find the shortest path from a source vertex.

**Lab 7: Dynamic Programming:**

1. Write a Program to implement 0/1 Knapsack problem.
2. Write a program to implement Chain matrix multiplications
3. Write a program to implement the Travelling Salesman Problem.

**Lab 8: Tree Algorithms: Traversal algorithms**

1. Write a program to implement Breadth First Search (BFS) algorithm.
2. Write a program to implement Depth First Search (DFS) algorithm.

**Lab 9: Approximation Algorithms**

1. Write a program to implement Vertex Cover problem.
2. Write a program to implement set cover problem.

**Lab 10: Backtracking Algorithms**

1. Write a program to implement 8 queens problem.

**Lab 11: Network Flow Algorithms**

1. Write a program to implement Ford-Fulkerson Algorithm.

| **COURSE OUTCOMES** |
|---|
| After completion of course, students would be able to: |
| 1. Acquire efficiency in programming languages through implementation of different classes of algorithms |
| 2. Acquire efficiency in using appropriate data structure for an algorithm to improve its overall efficiency |
| 3. Acquire knowledge to decide a suitable algorithm for a new, computationally hard problem |

**References:**

**Suggested books:**

1. Introduction to Algorithms, 4TH Edition, Thomas H Cormen, Charles E Lieserson, Ronald L Rivest and Clifford Stein, MIT Press/McGraw-Hill.
2. Fundamentals of Algorithms – E. Horowitz et al.
3. Data Structures and Algorithms using C, Fifth Edition, R. S. Salaria, Khanna Publishing.
4. C: The Complete Reference, Fourth Edition, Herbert Schildt, McGraw Hill Education.
5. Data Structures with C, First Edition, Seymour Lipschutz, McGraw Hill Education.

**Suggested reference books**

1. Algorithm Design, First Edition, Jon Kleinberg and ÉvaTardos, Pearson.
2. Algorithm Design: Foundations, Analysis, and Internet Examples, Second Edition,Michael T Goodrich and Roberto Tamassia, Wiley.
3. Algorithms -- A Creative Approach, Third Edition, UdiManber, Addison-Wesley, Reading, MA.
4. The C Programming Language, Second Edition, Brian W. Kernighan & Dennis Ritchie, Pearson Education India.
5. Data Structures Using C, Second Edition, ReemaTharej, Oxford.

- Mandatory Course, MC-401

**COMPUTER SCIENCE AND ENGINEERING**

## Semester-V (3rd Year)

**Software Engineering**
**Course Code: PCC-CS501**

| | |
|---|---|
| **Course Code** | **PCC-CS501** |
| **Course Name** | **Software Engineering** |
| **Credits** | 3 |
| **Pre-Requisites** | |

Software Engineering (SE) comprises the core principles consistent in software construction and maintenance: fundamental software processes and life-cycles, mathematical foundations of software engineering, requirements analysis, software engineering methodologies and standard notations, principles of software architecture and re-use, software quality frameworks and validation, software development, and maintenance environments and tools. An introduction to object-oriented software development process and design. Topics include: iterative development, interpretation of requirements and use case documents into code; application of design notation in UML and use of commonly-used design patterns. Current industry-strength programming languages, technologies and systems feature highly in the practical components, electives and projects of the course, but they are also taught with a view to understanding and applying principles underlying their more ephemeral character.

Course Objectives:

1. Knowledge of basic SW engineering methods and practices, and their appropriate application.
2. Describe software engineering layered technology and Process frame work.
3. A general understanding of software process models such as the waterfall and evolutionary models.
4. Understanding of software requirements and the SRS documents.
5. Understanding of the role of project management including planning, scheduling, risk management, etc.
6. Describe data models, object models, context models and behavioral models.
7. Understanding of different software architectural styles.

8. Understanding of implementation issues such as modularity and coding standards.
9. Understanding of approaches to verification and validation including static analysis, and reviews.
10. Understanding of software testing approaches such as unit testing and integration testing.
11. Describe software measurement and software risks.
12. Understanding of software evolution and related issues such as version management.
13. Understanding on quality control and how to ensure good quality software.

**Contracts:3L** **36 Hrs.**
**Credits- 3**

**Module I**

**Introduction, Software Lifecycle Models, Software Project Management [6L]**

Software Engineering –Objectives, Definitions, Feasibility Analysis, Software Process models - Waterfall Model, Prototype model, RAD, Evolutionary Models, Incremental, Spiral, Agile method

**Module II**

**Software Project Management [4L]**

Software Project Planning, Project Scheduling, Staffing, Cost-Benefit Analysis, COCOMO model.

**Module III**

**Software Requirements Engineering, Design [6L]**

Software Requirements Specification, Structured Analysis, Functional Design, Context diagram and DFD, Physical and Logical DFDs, Data Modelling, ER diagrams, Object Oriented Design, UML, Design Patterns.

**Module IV**

**Coding and Documentation [4L]**

Structured Programming, Modular Programming, Module Relationship- Coupling, Cohesion, OO Programming, Information Hiding, Reuse, System Documentation.

**Module V**

**Software Testing [8L]**

Testing Objectives, Types, Levels, Verification vs. Validation, Unit Testing, Integration Testing, Test Stub and Driver, Test Case, Test Suit design, System Testing

**Module V**

**Software Quality Assurance [5L]**

Software Quality Assurance, ISO 9001, SEI CMM, SIX SIGMA, Software Configuration Management.

**Module VI**

**Software Architecture[3L]**

Software Architectural Patterns

**Text Book:**

**1.** Software Engineering - A practitioner's approach– Pressman (TMH)

**Reference Books:**

1. Software Engineering- Pankaj Jalote (Wiley-India)
2. Software Engineering- Rajib Mall (PHI)
3. Software Engineering - Agarwal and Agarwal (PHI)
4. Ian Sommerville, Software Engineering, Addison-Wesley.

**Learning Outcomes:**

1. Basic knowledge and understanding of the analysis and design of complex systems.
2. Ability to apply software engineering principles and techniques.
3. Ability to develop, maintain and evaluate large-scale software systems.
4. To produce efficient, reliable, robust and cost-effective software solutions.
5. Ability to perform independent research and analysis.
6. To communicate and coordinate competently by listening, speaking, reading and writing English for technical and general purposes.
7. Ability to work as an effective member or leader of software engineering teams.
8. To manage time, processes and resources effectively by prioritizing competing demands to achieve personal and team goals Identify and analyzes the common threats in each domain.
9. Ability to understand and meet ethical standards and legal responsibilities.

**Database Management Systems**

**Subject Code:PCC-CS502**

| | | | |
|---|---|---|---|
| **YEAR- 3RD** | **SEMESTER- V** | **Professional Core Courses** | |
| **PCC-CS502** | **Database Management Systems** | **3L:0T:0P** | **3 credits** |
| **Pre-requisites** | | | |

**Objectives of the course:**

1. To understand the different issues involved in the design and implementation of a database system.
2. To study the physical and logical database designs, database modeling, relational, hierarchical, and network models
3. To understand and use data manipulation language to query, update, and manage a database.

4. To develop an understanding of essential DBMS concepts such as: database security, integrity, concurrency, distributed database, and intelligent database, Client/Server (Database Server), Data Warehousing.
5. To design and build a simple database system and demonstrate competence with the fundamental tasks involved with modeling, designing, and implementing a DBMS.

**Detailed contents**

Module 1: [ 9L]

**Database system architecture:** Data Abstraction, Data Independence, Data Definition Language (DDL), Data Manipulation Language (DML).
**Data models:** Entity-relationship model, network model, relational and object oriented data models, integrity constraints, data manipulation operations.

**Module 2: [13L]**
**Relational query languages:** Relational algebra, Tuple and domain relational calculus, SQL, DDL and DML constructs, Open source and Commercial DBMS - MYSQL, ORACLE, DB2, SQL server.
**Relational database design:** Domain and data dependency, Armstrong's axioms, Normal forms, Dependency preservation, Lossless design.
**Query processing and optimization:** Evaluation of relational algebra expressions, Query equivalence, Join strategies, Query optimization algorithms.

**Module 3: [3L]**
**Storage strategies**: Indices, B-trees, hashing.

**Module 4: [5L]**
**Transaction processing:** Concurrency control, ACID property, Serializability of scheduling, Locking and timestamp based schedulers, Multi-version and optimistic Concurrency Control schemes, Database recovery.

**Module 5: [3L]**
**Database Security:** Authentication, Authorization and access control, DAC, MAC and RBAC models, Intrusion detection, SQL injection.

**Module 6: [3L]**
**Advanced topics:** Object oriented and object relational databases, Logical databases, Web databases, Distributed databases, Data warehousing and data mining.

**Text books:**
1. “Database System Concepts”, 6th Edition by Abraham Silberschatz, Henry F. Korth, S. Sudarshan, McGraw-Hill.

2. “Fundamentals of Database Systems”, 5th Edition by R. Elmasri and S. Navathe, Pearson Education
3. “An Introduction to Database Systems”, by C.J. Date, A. Kannan, S. Swamynathan., Pearson Education

**Reference books**

1. “Principles of Database and Knowledge – Base Systems”, Vol 1 by J. D. Ullman, Computer Science Press.
2. “Foundations of Databases”, Reprint by Serge Abiteboul, Richard Hull, Victor Vianu, Addison-Wesley
3. “Database Management System”, by Ramakrishnan. McGraw-Hill

**Course Outcomes:**

**On completion of the course students will be able to**

1. For a given problem (requirement specification) design the databases using E-R diagram method and convert it into Relational Database.
2. For a given query write relational algebra and Relational Calculus expressions for that query and optimize the developed expressions.
3. For a given specification construct the SQL queries for Open source and Commercial DBMS.
4. For a given query optimize its execution using Query optimization algorithms.
5. Understand and implement transaction processing, concurrency control and Recovery system.
6. Understand different type of advanced database and know about database security.

## Formal Language & Automata Theory

**Subject Code: PCC-CS-503**

| | |
|---|---|
| Course Code | PCC-CS503 |
| Course Name | Formal Language & Automata Theory |
| Credits | 3 |
| L-T-P | 3-0-0 |
| Pre-Requisites | |

**Total Number of Lectures: 36**

| Course Objective |
|---|
| The aim of the course is to<br>1. Develop a formal notation for strings, languages and machines<br>2. Design finite automata to accept a set of string of a language<br>3. Prove that a given language is regular and apply the closure properties of languages.<br>4. Design context free grammars to generate strings from a context free language and convert them into normal forms. |

| |
|---|
| 5. Prove equivalence of languages accepted by push down automata and languages generated by context free grammars.<br>6. Identify the hierarchy of formal languages, grammars and machines.<br>7. Distinguish between computability and non-computability, decidability and undecidability. |

| **LECTURES WITH BREAKUP** | **NO. OF LECTURES** |
|---|---|
| **Unit 1:**<br>Introduction: Alphabet, languages and grammars, productions and derivation, Chomsky hierarchy of languages. | **4** |
| **Unit 2:**<br>Regular languages and finite automata: Regular expressions and languages, deterministic finite automata (DFA) and equivalence with regular expressions, nondeterministic finite automata (NFA) and equivalence with DFA, regular grammars and equivalence with finite automata, properties of regular languages, pumping lemma for regular languages, minimization of finite automata. | **8** |
| **Unit 3:**<br>Context-free languages and pushdown automata: Context-free grammars (CFG) and languages (CFL), Chomsky and Greibach normal forms, nondeterministic pushdown automata (NPDA) and equivalence with CFG, parse trees, ambiguity in CFG, pumping lemma for context-free languages, deterministic pushdown automata(DPDA), closure properties of CFLs. | **8** |
| **Unit 4:**<br>Context-sensitive languages: Context-sensitive grammars (CSG) and languages, linear bounded automata (LBA) and equivalence with CSG. | **4** |
| **Unit 5:**<br>Turing machines: The basic model for Turing machines (TM), Turing recognizable (recursively enumerable) and Turing-decidable (recursive) languages and their closure properties, variants of Turing machines, nondeterministic TMs and equivalence with deterministic TMs, unrestricted grammars and equivalence with Turing machines, TMs as enumerators. | **7** |
| **Unit 6:**<br>Undecidability: Church-Turing thesis, universal Turing machine, the universal and diagonalization languages, reduction between languages and Rice s theorem, undecidable problems about languages. | **5** |

| **Course Outcomes** |
|---|
| **On completion of the course, students will be able to:** |
| • Write a formal notation for strings, languages and machines. |

| |
|---|
| • Design finite automata to accept a set of strings of a language. |
| • For a given language determine whether the given language is regular or not. |
| • Design context free grammars to generate strings of context free language. |
| • Determine equivalence of languages accepted by Push Down Automata and languages generated by context free grammars |
| • Write the hierarchy of formal languages, grammars and machines. |
| • Distinguish between computability and non-computability and Decidability and undecidability. |

**Text Books / References Books:**

1. John E. Hopcroft, Rajeev Motwani and Jeffrey D. Ullman, Introduction to Automata Theory, Languages, and Computation, Pearson Education Asia.
2. Harry R. Lewis and Christos H. Papadimitriou, Elements of the Theory of Computation, Pearson Education Asia.
3. Dexter C. Kozen, Automata and Computability, Undergraduate Texts in Computer Science, Springer.
4. Michael Sipser, Introduction to the Theory of Computation, PWS Publishing.
5. John Martin, Introduction to Languages and The Theory of Computation, TataMcGraw Hill.

**Object Oriented Programming**
**PCC-CS504**
**2L:0T:0 P 2 Credits**
**Total Number of Lectures: 24**

| | |
|---|---|
| **Course Code** | PCC-CS504 |
| **Course Name** | Object Oriented Programming |
| **Credits** | 2L:0T:0P 2 Credits |
| **Pre-Requisites** | PCC-CS302 : Data structure & Algorithms |

**Total Number of Lectures: 24**

| **COURSE OBJECTIVES** |
|---|
| **1.** To develop conceptual understanding of Object Oriented System. |
| **2.** To understand how a real world problem can be mapped to object oriented problem domain. |
| **3.** To solve different industry level problems & to learn its applications. |

**Syllabus:**

**Module 1:**

**Object oriented design**

Concepts of object oriented programming language, Major and minor elements, Object, Class, relationships among objects, aggregation, links, relationships among classes-association, aggregation, using, instantiation, meta-class, grouping constructs.

**Object oriented concepts**

Difference between OOP and other conventional programming – advantages and disadvantages. Class, object, message passing, inheritance, encapsulation, polymorphism

**Module 2:**

**Language features to be covered: Class & Object proprieties**

Basic concepts of java programming – advantages of java, byte-code & JVM, data types, operators, control statements & loops, creation of class, object, constructor, finalize and garbage collection, use of method overloading, this keyword, use of objects as parameter & methods returning objects, call by value & call by reference, static variables & methods, Arrays, Strings, Vectors and Wrapper classes, command line arguments, basics of I/O operations – keyboard input using BufferedReader & Scanner classes.

**Module 3:**

**Reusability properties** – Super class & subclasses including multilevel hierarchy, process of constructor chaining in inheritance, use of super and final keywords with super() method, dynamic method dispatch, Final variables and methods, Final classes, use of abstract classes & methods, access visibility, interfaces and multiple inheritance – defining, extending and implementing interfaces.

**Module 4:**

**Packages & Exception handling –** Using System packages, naming conventions,

Creation of packages, adding classes to a Package, importing packages, member access for packages, Static import.

Exception handling basics, types of Errors, different types of Exception classes, use of try & catch with throw, throws & finally, creation of user defined exception classes.

**Module 5:**

**Multithreading** : Basics of multithreading, main thread, thread life cycle, creation of multiple threads, thread priorities, thread synchronization, inter- thread communication, deadlocks for threads, suspending & resuming threads.

**Module 6:**

**Applet Programming** – Basics of applet programming, difference between application & applet programming, applet life cycle, building applet code, Web page design, Applet tag, parameter passing in applets, Input from users, Event handling

| **LECTURE WITH BREAKUP** | **NO. OF LECTURES** |
|---|---|
| **Module 1:**<br><br>**Object oriented design**<br><br>Concepts of object oriented programming language, Major and minor elements, Object, Class, relationships among objects, aggregation, links, relationships among classes-association, aggregation, using, instantiation, meta-class, grouping constructs.<br><br>**Object oriented concepts**<br><br>Difference between OOP and other conventional programming – advantages and disadvantages. Class, object, message passing, inheritance, encapsulation, polymorphism | **[4L]** |

| | |
|---|---|
| **Module 2:**<br><br>**Language features to be covered: Class & Object proprieties**<br><br>Basic concepts of java programming – advantages of java, byte-code & JVM, data types, operators, control statements & loops, creation of class, object, constructor, finalize and garbage collection, use of method overloading, this keyword, use of objects as parameter & methods returning objects, call by value & call by reference, static variables & methods, Arrays, Strings, Vectors and Wrapper classes, command line arguments, basics of I/O operations – keyboard input using BufferedReader & Scanner classes. | **[4L]** |
| **Module 3:**<br><br>**Reusability properties** – Super class & subclasses including multilevel hierarchy, process of constructor chaining in inheritance, use of super and final keywords with super() method, dynamic method dispatch, Final variables and methods, Final classes, use of abstract classes & methods, access visibility, interfaces and multiple inheritance – defining, extending and implementing interfaces. | **[4L]** |
| **Module 4:**<br><br>**Packages & Exception handling –** Using System packages, naming conventions, Creation of packages, adding classes to a Package, importing packages, member access for packages, Static import.<br><br>Exception handling basics, types of Errors, different types of Exception classes, use of try & catch with throw, throws & finally, creation of user defined exception classes. | **[4L]** |
| **Module 5:**<br><br>**Multithreading** : Basics of multithreading, main thread, thread life cycle, creation of multiple threads, thread priorities, thread synchronization, inter- thread communication, deadlocks for threads, suspending& resuming threads. | **[4L]** |
| **Module 6:**<br><br>**Applet Programming** – Basics of applet programming, difference between application & applet programming, applet life cycle, building | **[4L]** |

| | |
|---|---|
| applet code, Web page design, Applet tag, parameter passing in applets, Input from users, Event handling | |

| **COURSE OUTCOMES** |
|---|
| **After completion of course, students would be able to:** |
| CO1: Visualize a given problem scenario in terms of classes and objects. |
| CO2: Acquire the knowledge about different types of inheritance & polymorphism, interface, package, vector and wrapper class. |
| CO3: Apply object oriented programming concepts through Java for problem solving. |
| **CO4: Acquire knowledge about threads, thread synchronization and applets and their life cycle.** |

**Textbooks/References:**


1. Rambaugh, James Michael, Blaha – "Object Oriented Modelling and Design" – Prentice Hall, India
2. Ali Bahrami – "Object Oriented System Development" – Mc Graw Hill
3. Rambaugh, James Michael, Blaha – "Object Oriented Modelling and Design" – Prentice Hall, India
4. Ali Bahrami – "Object Oriented System Development" – Mc Graw Hill
5. Patrick Naughton, Herbert Schildt – "The complete reference-Java2" – TMH
6. R.K Das – "Core Java For Beginners" – VIKAS PUBLISHING
7. Deitel and Deitel – "Java How to Program" – 6th Ed. – Pearson
8. Ivor Horton's Beginning Java 2 SDK – Wrox
9. E. Balagurusamy – " Programming With Java: A Primer" – TMH

**Humanities II**
**Subject Code: HSMC-501**
**Credit: 3**

| | |
|---|---|
| **Course Code** | **HSMC-501** |
| **Course Name** | **Humanities II** |
| **Credits** | **3L:0T:0P 3 Credits** |
| **Pre-Requisites** | |

**Objective:** the objective of this course is to identify and understand the importance and significance of human factors in the organization. It also illuminates the philosophy, tools and relevance of human resource development and its role for individual and organizational growth.

| Sl. | Contents | **Lecture Hour** |
|---|---|---|
| 1. | HRD – Concept, Definition, Importance of HRD, Origin & Need of HRD, Human Development & HRD, Different approaches of HRD, HRD as a system & subsystem, Attributes of HRD Managers, HRD in Indian Industry | 10 |
| 2. | Recruitment: Meaning, Sources of Recruitment, Recruitment, Placement and Induction Process, Promotion, Demotion and Transfer.<br>Selection: Concept, Policy, Process, Different Types of Selection Tests and its Application. | 08 |
| 3. | Concept of HR Training & Development, Importance & Need of Training, Designing & Evaluating Training & Development Programs, Coaching & Mentoring Responsibilities of a Training Manager, What are the Challenges, A Training Manager Faces. Different Types of Training; On the Job Training, Off the Job Training' Talent Management & Career Development. | 12 |
| 4. | HRD Audit & Evaluation; Concept & Definition, Purpose of HRD Evaluation Programs, Steps of HRD Evaluation, Models & Framework of Evaluation, Role of HRD Audit in Companies, Methods and its limitations, Balance Scorecard Approach, HRD Scorecard Approach, appreciative enquiry. | 10 |

| | | |
|---|---|---|
| 5. | HRD Climate and Culture in the Organization, Organizational Communication & Motivation, Qualities of HRD Managers, HRD & Organizational Development, HRD in Indian Industry, OCTAPACE, HRD Practices in different Organizations Government, Private & MNCs, Global HRD Scenario | 08 |
| | Total | 48 |

Suggested Reading


1. Ashwathappa.K (2005). Human Resource and Personnel Management. New Delhi: Tata McGraw Hill Publishers.
2. Wilson, J.P. (2005). Human Resource Development – Learning and Training for individuals and organizations. London: Kogan page
3. Deb. J. (2012). Human Resource Development – Theory and Practice. New Delhi: Ane Books P Ltd.
4. Werner, J.M. and Desimone, R. L. (2006). Human Resource Development. New Delhi: Cengage Learning.
5. Rao,T.V. (2010). Performance management and appraisal systems – HR tools for global competitiveness. New Delhi: Response Books.


**Database Management Systems LAB**
**Subject Code: PCC-CS592**

| **YEAR- 3RD** | **SEMESTER- V** | | |
|---|---|---|---|
| **PCC-CS592** | **Database Management Systems LAB** | **0L:0T:4P** | **2credits** |
| **Pre-requisites** | | | |

**Detailed contents:**

**Experiments should include but not limited to :**

**Structured Query Language**

1. Creating Database
   - Creating a Database
   - Creating a Table
   - Specifying Relational Data Types
   - Specifying Constraints
   - Creating Indexes
2. Table and Record Handling
   - INSERT statement
   - Using SELECT and INSERT together

DELETE, UPDATE, TRUNCATE statements
DROP, ALTER statements

3. Retrieving Data from a Database
   The SELECT statement
   Using the WHERE clause
   Using Logical Operators in the WHERE clause
   Using IN, BETWEEN, LIKE , ORDER BY, GROUP BY and HAVING Clause
      Using Aggregate Functions
      Combining Tables Using JOINS
      Subqueries
4. Database Management
   Creating Views
   Creating Column Aliases
   Creating Database Users
   Using GRANT and REVOKE

5. PL/SQL
   Cursors in Oracle PL / SQL
   Writing Oracle PL / SQL Stored Procedures

**Object Oriented Programming Lab**
**PCC-CS594**
**0L:0T:4 P 2 Credits**
**Total Number of Lectures: 48**

| **Course Code** | **PCC-CS594** |
|---|---|
| **Course Name** | **Object Oriented Programming Lab** |
| **Credits** | **0L:0T:4P 2 Credits** |
| **Pre-Requisites** | **PCC-CS302 : Data structure & Algorithms** |

**Total Number of Lectures: 48**

| **COURSE OBJECTIVES** |
|---|
| **4. To develop conceptual understanding of Object Oriented System.** |
| **5. To understand how a real world problem can be mapped to object oriented problem domain.** |
| **6. To solve different industry level problems & to learn its applications.** |

**Syllabus:**

Simple Classes for understanding objects, member functions and Constructors –

Inheritance – Compile time Polymorphism - Run time Polymorphism- Understanding of wrapper class, arrays - Assignments on developing interfaces- multiple inheritance, extending interfaces - exception handling mechanism in java - creating and accessing packages - multithreaded programming - applet programming

| COURSE OUTCOMES |
|---|
| **After completion of course, students would be able to:** |
| **CO1: Understand the object oriented approach of software development.** |
| **CO2: Learn about proper object oriented design principles while focussing on the reusability concept.** |
| **CO3: Implement a given design using Java.** |

**Textbooks/References:**


1. Rambaugh, James Michael, Blaha – "Object Oriented Modelling and Design" – Prentice Hall, India
2. Ali Bahrami – "Object Oriented System Development" – Mc Graw Hill
3. Patrick Naughton, Herbert Schildt – "The complete reference-Java2" – TMH
4. R.K Das – "Core Java For Beginners" – VIKAS PUBLISHING
5. Deitel and Deitel – "Java How to Program" – 6th Ed. – Pearson
6. Ivor Horton's Beginning Java 2 SDK – Wrox
7. E. Balagurusamy – " Programming With Java: A Primer" – TMH


**Graph Theory**
**PEC-CS501A**

| | |
|---|---|
| Course Code | PEC-CS501A |
| Course Name | Graph Theory |
| Credits | 3 |
| L-T-P | 3-0-0 |
| Pre-Requisites | Digital Logic |

**Total Number of Lectures: 36**

| Course Objective |
|---|
| i) To understand and apply the fundamental concepts in graph theory<br>ii) To apply graph theory based tools in solving practical problems |

| | |
|---|---|
| iii) To improve the proof writing skills.<br>iv) Be familiar with the most fundamental Graph Theory topics and results.<br>v) Be exposed to the techniques of proofs and analysis. | |
| **LECTURES WITH BREAKUP** | **NO. OF LECTURES** |
| **Unit 1:** GRAPHS AND SUBGRAPH - Graphs and Simple Graphs, Graph Isomorphism, The Incidence and Adjacency Matrices, Subgraphs, Vertex Degrees, Paths and Connection, Cycles. | 6 |
| **Unit 2:** TREES, CONNECTIVITY & PLANARITY<br>Trees (properties, distances and centroids), vertex and edge connectivity,Spanning trees – Fundamental circuits – Spanning trees in a weighted graph – cut sets – Properties of cut set – All cut sets – Fundamental circuits and cut sets – Connectivity and separability – Network flows – 1-Isomorphism – 2-Isomorphism – Combinational and geometric graphs – Planer graphs – Different representation of a planer graph, Matrix Representation of Graphs ( Adjacency and Incidence Matrices) | 7 |
| **Unit 3**: MATRICES, COLOURING AND DIRECTED GRAPH Chromatic number – Chromatic partitioning – Chromatic polynomial – Matching – Covering – Four color problem – Directed graphs – Types of directed graphs – Digraphs and binary relations – Directed paths and connectedness – Euler graphs. Applications - A Storage Problem | 8 |
| **Unit 4:**<br>NETWORKS - Flows , Cuts, The Max-Flow Min-Cut Theorem, Ford-Fulkerson Algorithm for Maximum Flow ; Floyd Algorithm ; Max–Flow and Min–Cut Theorem, Applications - Menger's Theorems, Feasible Flows | 8 |
| **Unit 5:**<br>**Graph Algorithms:** BFS, DFS, connected components, topological sort, Minimal Spanning Tree : Kruskal's Algorithm ; Prim's Algorithm, shortest paths - single source and all pairs. | 7 |
| **Course Outcomes** | |
| **After completion of the course, students would be able to:** | |
| ▪ know some important classes of graph theoretic problems; | |

| |
|---|
| ▪ be able to formulate and prove central theorems about trees, matching, connectivity, colouring and planar graphs; |
| ▪ be able to describe and apply some basic algorithms for graphs;<br><br>● The students will be able to apply principles and concepts of graph theory in practical situations<br><br>● Solve problems using basic graph theory<br>● Identify induced subgraphs, cliques, matchings, covers in graphs<br>● Determine whether graphs are Hamiltonian and/or Eulerian<br>● Solve problems involving vertex and edge connectivity, planarity and crossing numbers<br>● Solve problems involving vertex and edge coloring<br>● Model real world problems using graph theory |

**References:**


1. Narsingh Deo, "Graph Theory: With Application to Engineering and Computer Science", Prentice Hall of India, 2003.
2. Grimaldi R.P. "Discrete and Combinatorial Mathematics: An Applied Introduction", Addison Wesley, 1994.
3. Clark J. and Holton D.A, "A First Look at Graph Theory", Allied Publishers, 1995.
4. Gibbons, Algorithmic Graph Theory, Cambridge University Press.


**Artificial Intelligence**
**PEC-CS501C**

| | |
|---|---|
| **Course Code** | PEC-CS501C |
| **Course Name** | Artificial Intelligence |
| **Credits** | 3 |
| **Pre-Requisites** | Basic Mathematics, Computer programming |

**COURSE OBJECTIVES** Total Number of Lectures: 36

| |
|---|
| • To provide a strong foundation of fundamental concepts in Artificial Intelligence. |
| • To provide a basic exposition to the goals and methods of Artificial Intelligence. |
| • To enable the student to apply these techniques in applications which involve perception, reasoning and learning. |

| LECTURE WITH BREAKUP | NO.OF LECTURS |
|---|---|
| **Introduction**<br>Overview of Artificial intelligence- Problems of AI, AI technique, Tic - Tac - Toe problem. | 2 |
| **Intelligent Agents**<br>Agents & environment, nature of environment, structure of agents, goal based agents, utility based agents, learning agents. | 2 |
| **Problem Solving**<br>Problems, Problem Space & search: Defining the problem as state space search, production system, problem characteristics, issues in the design of search programs. | 2 |
| **Search techniques**<br>Solving problems by searching :problem solving agents, searching for solutions; uniform search strategies: breadth first search, depth first search, depth limited search, bidirectional search, comparing uniform search strategies. | 3 |
| **Heuristic search strategies**<br>Greedy best-first search, A* search, memory bounded heuristic search: local search algorithms & optimization problems: Hill climbing search, simulated annealing search, local beam search, genetic algorithms; constraint satisfaction problems, local search for constraint satisfaction problems. | 4 |
| **Adversarial search**<br>Games, optimal decisions & strategies in games, the minimax search procedure, alpha-beta pruning, additional refinements, iterative deepening. | 3 |
| **Knowledge & reasoning**<br>Knowledge representation issues, representation & mapping, approaches to knowledge representation, issues in knowledge representation. | 3 |
| **Using predicate logic**<br>Representing simple fact in logic, representing instant & ISA relationship, computable functions & predicates, resolution, natural deduction. | 2 |
| **Representing knowledge using rules**<br>Procedural verses declarative knowledge, logic programming, forward verses backward reasoning, matching, control knowledge. | 3 |
| **Probabilistic reasoning**<br>Representing knowledge in an uncertain domain, the semantics of Bayesian networks, Dempster-Shafer theory, Fuzzy sets & fuzzy logics. | 3 |
| **Planning** | 2 |

| | |
|---|---|
| Overview, components of a planning system, Goal stack planning, Hierarchical planning, other planning techniques. | |
| **Natural Language processing**<br>Introduction, Syntactic processing, semantic analysis, discourse & pragmatic processing. | 2 |
| **Learning**<br>Forms of learning, inductive learning, learning decision trees, explanation based learning, learning using relevance information, neural net learning & genetic learning. | 3 |
| **Expert Systems**<br>Representing and using domain knowledge, expert system shells, knowledge acquisition. | 2 |

**COURSE OUTCOMES**

After completion of course, students would be able to:

| |
|---|
| 1. Understand the various searching techniques, constraint satisfaction problem and example problems- game playing techniques. |
| 2. Apply these techniques in applications which involve perception, reasoning and learning. |
| 3. Explain the role of agents and how it is related to environment and the way of evaluating it and how agents can act by establishing goals. |
| 4. Acquire the knowledge of real world Knowledge representation. |

**References:**


1. Artificial Intelligence A Modern Approach, Stuart Russel Peter Norvig Pearson
2. Artificial Intelligence, Ritch & Knight, TMH
3. Introduction to Artificial Intelligence & Expert Systems, Patterson, PHI
4. Poole, Computational Intelligence, OUP
5. Expert Systems, Giarranto, VIKAS


**Image Processing**

**PEC-CS501D**

| | |
|---|---|
| **Course Code** | PEC-CS501D |
| **Course Name** | Image Processing |
| **Credits** | 3 |
| **Pre-Requisites** | Basic Mathematics |

**COURSE OBJECTIVES**

Total Number of Lectures: 36

| |
|---|
| • To study the image fundamentals and mathematical transforms necessary for image processing. |
| • To study the image enhancement techniques |
| • To study image restoration procedures. |

| LECTURE WITH BREAKUP | NO.OF LECTURES |
|---|---|
| **Introduction**<br>Background, Digital Image Representation, Fundamental steps in Image Processing, Elements of Digital Image Processing – Image Acquisition, Storage, Processing, Communication, Display. | 3 |
| **Digital Image Formation**<br>A Simple Image Model, Geometric Model- Basic Transformation (Translation, Scaling, Rotation), Perspective Projection, Sampling & Quantization - Uniform & Non uniform. | 4 |
| **Mathematical Preliminaries**<br>Neighbour of pixels, Connectivity, Relations, Equivalence & Transitive Closure; Distance Measures, Arithmetic/Logic Operations, Fourier Transformation, Properties of The Two Dimensional Fourier Transform, Discrete Fourier Transform, Discrete Cosine & Sine Transform. | 8 |
| **Image Enhancement**<br>Spatial Domain Method, Frequency Domain Method, Contrast Enhancement -Linear & Nonlinear Stretching, Histogram Processing; Smoothing - Image Averaging, Mean Filter, Low-pass Filtering; Image Sharpening. High-pass Filtering, High-boost Filtering, Derivative Filtering, Homomorphic Filtering; Enhancement in the frequency domain - Low pass filtering, High pass filtering. | 8 |
| **Image Restoration**<br>Degradation Model, Discrete Formulation, Algebraic Approach to Restoration - Unconstrained & Constrained; Constrained Least Square Restoration, Restoration by Homomorphic Filtering, Geometric Transformation - Spatial Transformation, Gray Level Interpolation. | 6 |
| **Image Segmentation**<br>Point Detection, Line Detection, Edge detection, Combined detection, Edge Linking & Boundary Detection - Local Processing, Global Processing via The Hough Transform; Thresholding - Foundation, Simple Global Thresholding, Optimal Thresholding; Region Oriented | 7 |

| | |
|---|---|
| Segmentation - Basic Formulation, Region Growing by Pixel Aggregation, Region Splitting & Merging. | |

**COURSE OUTCOMES**

After completion of course, students would be able to:

| |
|---|
| 5. Understand the need for image transforms different types of image transforms and their properties. |
| 6. Develop different types of image processing application. |
| 7. Learn different techniques employed for the enhancement of images. |
| 8. Learn different causes for image degradation and overview of image restoration techniques. |
| 9. Learn different feature extraction techniques for image analysis and recognition. |

**References:**


1. Digital Image Processing, Gonzalves,Pearson
2. Digital Image Processing, Jahne, Springer India
3. Digital Image Processing & Analysis,Chanda & Majumder,PHI
4. Fundamentals of Digital Image Processing, Jain, PHI
5. Image Processing, Analysis & Machine Vision, Sonka, VIKAS
6. Getting Started with GIS- Clarke Keith. C; PE.
7. Concepts & Techniques of GIS - Lo C.P, Albert, Yeung K.W- PHI.


**Soft Computing**
**PEC-CS501E**

| | |
|---|---|
| **Course Code** | PEC-CS501E |
| **Course Name** | Soft Computing |
| **Credits** | 3 |
| **Pre-Requisites** | Basic knowledge of mathematics |

Total Number of Lectures:36

| **COURSE OBJECTIVE** |
|---|
| • To introduce basic concepts of soft computing techniques with illustrative examples. |
| • To give students knowledge in neural networks, fuzzy sets, fuzzy logic, genetic algorithms and other meta-heuristics |
| • To provide students an hand-on experience on MATLAB/Python to implement various strategies. |

| LECTURE WITH BREAKUP | NO. OF LECTURES |
|---|---|
| **Unit 1**<br>**INTRODUCTION TO SOFT COMPUTING:** Evolution ofComputing: Basic Soft Computing Constituents: Artificial Neural Networks, Meta-heuristic search, Fuzzy Systems | 3 |
| **Unit 2**<br>**FUZZY SETS:** Fuzzy Sets and crisp sets, Operations on Fuzzy Sets, Membership Functions, Fuzzy Relations:<br>**FUZZY SYSTEM:** Fuzzy Rules and Fuzzy Reasoning, Fuzzy Inference Systems, Fuzzy Decision Making. | 8 |
| **Unit 3**<br>**NEURAL NETWORKS:** Introduction, benefits of neural networks, basic models of neuron- MP neuron, Perceptron, Architecture of neuron- Single layer, Multi layer Feed forward Networks, Adaptive Networks, Different learning strategies of neurons-Supervised Learning Neural Networks, RadialBasis Function Networks : Unsupervised Learning Neural Networks, MAX Net, SOM, Advances in Neural networks | 9 |
| **Unit 4**<br>**Meta-heuristics:** Introduction to Meta-heuristic search: Simulated Annealing (SA), Genetic Algorithms (GA), Particle Swarm Optimization (PSO), Applications to real world optimization problems | 6 |
| **Unit 5**<br>**Matlab/Python Lib:** Introduction to Matlab/Python, Arrays and array operations, Functions and Files, Study of neural network toolbox and fuzzy logic toolbox, Simple implementation of Artificial Neural Network and Fuzzy Logic | 7 |
| **Unit 6**<br>Recent Trands in deep learning, various classifiers, neural networks and genetic algorithm.Implementation of recently proposed soft computing techniques. | 3 |

| COURSE OUTCOMES |
|---|
| After completion of course, students would be able to: |
| • Identify and describe soft computing techniques, like ANN, Fuzzy Systems, |

| |
|---|
| Meta-heuristics |
| • Handle real world decision and optimization problems using soft computing techniques |
| • Apply soft computing to handle uncertainty, imprecision and solve various engineering problems |

References


1. Jyh:Shing Roger Jang, Chuen:Tsai Sun, EijiMizutani, Neuro-Fuzzy and Soft Computing®, Prentice-Hall of India, 2003.
2. George J. Klir and Bo Yuan, Fuzzy Sets and Fuzzy Logic-Theory and Applications®, Prentice Hall, 1995.
3. Roy S., Chakraborty U, Introduction to Soft Computing: Neuro-Fuzzy and Genetic Algorithms, Pearson, 2013.
4. El-Ghazali Talbi, Meta-heuristics: From Design to Implementation, Wiley, 2009.
5. John Yen, Reza Langari, Fuzzy Logic: Intelligence, Control and Information, Pearson, 2007.
6. MATLAB Toolkit Manual


| |
|---|
| problems. |
| • Apply genetic algorithms to combinatorial optimization problems. |
| • Evaluate and compare solutions by various soft computing approaches for a given problem. |

References


Jyh:Shing Roger Jang, Chuen:Tsai Sun, EijiMizutani, Neuro:Fuzzy and Soft Computing®, Prentice:Hall of India, 2003.

George J. Klir and Bo Yuan, Fuzzy Sets and Fuzzy Logic:Theory and Applications®, Prentice Hall, 1995.

MATLAB Toolkit Manual


- Mandatory Course, MC501, Constitution of India

**COMPUTER SCIENCE AND ENGINEERING**

**Semester-VI (3rd Year)**

**Compiler Design**
**PCC-CS601**

| | |
|---|---|
| Course Code | PCC-CS601 |
| Course Name | Compiler Design |
| Credits | 3 |
| L-T-P | 3-0-0 |

**Total Number of Lectures: 36**

| Course Objective | |
|---|---|
| The aim of the course is to<br>1. Understand and list the different stages in the process of compilation.<br>2. Identify different methods of lexical analysis<br>3. Design top-down and bottom-up parsers<br>4. Identify synthesized and inherited attributes<br>5. Develop syntax directed translation schemes<br>6. Develop algorithms to generate code for a target machine | |
| **LECTURES WITH BREAKUP** | **NO. OF LECTURES** |
| **Unit 1:**<br>Introduction: Phases of compilation and overview. Cousins of compiler. | **3** |
| **Unit 2:**<br>Lexical Analysis : Regular languages, finite automata, regular expressions, regular expressions to NFA, regular expression to DFA, | **5** |

| | |
|---|---|
| scanner generator (Lex). | |
| **Unit 3:**<br>Syntax Analysis : Context-free languages and grammars, push-down automata, LL(1) grammars and top-down parsing, operator grammars, LR(0), SLR(1), LR(1), LALR(1) grammars and bottom-up parsing, ambiguity and LR parsing, LALR(1) parser generator (YACC) | **6** |
| **Unit 4:**<br>Semantic Analysis: Attribute grammars, syntax directed definition, evaluation and flow of attribute in a syntax tree. | **4** |
| **Unit 5:**<br>Symbol Table: Its structure, symbol attributes and management. | **2** |
| **Unit 6:**<br>Run-time environment: Procedure activation, parameter passing, value return, memory allocation, and scope. | **3** |
| **Unit 7:**<br>Intermediate Code Generation: Translation of different language features, different types of intermediate forms. | **3** |
| **Unit 8:**<br>Optimization): Analysis - control-flow, data-flow dependence etc.; Code improvement - local optimization, global optimization, loop optimization, peep-hole optimization etc. Architecture dependent code improvement - instruction scheduling (for pipeline), loop optimization (for cache memory) etc. | **5** |
| **Unit 9:**<br>Code generation: Issues, a simple code generator, Register allocation and target code generation<br>Advanced topics: Type systems, data abstraction, compilation of Object Oriented features and non-imperative programming languages. | **5** |
| **Course Outcomes** | |
| **After completion of the course, students would be able to:** | |
| • For a given grammar specification develop the lexical analyser | |
| • For a given parser specification design top-down and bottom-up parsers | |
| • Develop syntax directed translation schemes | |
| • Develop algorithms to generate code for a target machine | |

**References:**


1. Aho, Sethi, Ullman - “Compiler Principles, Techniques and Tools” - Pearson Education.
2. Holub - “Compiler Design in C” - PHI.
3. Keith D. Cooper & Linda Torczon – Engineering a Compiler – Morgan Kaufmann Publications, 2nd Ed.


**Computer Networks**
**PCC-CS602**

| YEAR- 3RD | | SEMESTER- VI | |
|---|---|---|---|
| PCC-CS602 | Computer Networks | 3L:0T: 0P | 3 credits |

**Objectives of the course**

1. To develop an understanding of modern network architectures from a design and performance perspective.
2. To introduce the student to the major concepts involved in wide-area networks (WANs).
3. Local area networks (LANs) and Wireless LANs (WLANs).
4. To provide an opportunity to do network programming
5. To provide a WLAN measurement ideas.

**Detailed contents:**

**Module I [ 4L]**

**Data communication Components**

Representation of data and its flow Networks , Various Connection Topology, Protocols and Standards, OSI model, Transmission Media, LAN: Wired LAN, Wireless LANs, Connecting LAN and Virtual LAN, Techniques for Bandwidthutilization: Multiplexing - Frequency division, Time division and Wave division, Conceptson spread spectrum.

**Module II [10L]**

**Data Link Layer and Medium Access Sub Layer**

Error Detection and Error Correction -Fundamentals, Block coding, Hamming Distance, CRC; Flow Control and Error controlprotocols - Stop and Wait, Go back – N ARQ, Selective Repeat ARQ, Sliding Window,Piggybacking, Random Access, Multiple access protocols -Pure ALOHA,Slotted ALOHA, CSMA/CD,CDMA/CA

**Module III: [8L]**

**Network Layer:** Internetworking & devices, Logical addressing – IPV4, IPV6; Address mapping- ARP, RARP, BOOTP and DHCP, Routing: static vs. dynamic routing, Unicast Routing protocols: RIP, OSPF, BGP; Other Protocols: IP, ICMP.

**Module IV: [5L]**

**Transport Layer:** Process to Process Communication, User Datagram Protocol (UDP), Transmission Control Protocol (TCP), SCTP Congestion Control; Quality of Service, QoS improving techniques: Leaky Bucket and Token Bucket algorithm.

**Module V: [ 8L]**

**Application Layer:** Introduction toDNS, DDNS, TELNET, SMTP, FTP, WWW, HTTP, SNMP.

**Module VI:**
**Advance Topics [8L]**
Bluetooth, Mobility and Security in Computer Network, Internet of Things, Basic Cryptography Concepts, Firewalls etc.

**Text books:**
1. Data Communication and Networking, 4th Edition, Behrouz A. Forouzan, McGraw-Hill.
2. Data and Computer Communication, 8th Edition, William Stallings, Pearson Prentice Hall India.

**Reference books:**
1. Computer Networks, 8th Edition, Andrew S. Tanenbaum, Pearson New International Edition.
2. Internetworking with TCP/IP, Volume 1, 6th Edition Douglas Comer, Prentice Hall of India.
3. TCP/IP Illustrated, Volume 1, W. Richard Stevens, Addison-Wesley, United States of America.

**Course Outcomes:**

1. Explain the functions of the different layer of the OSI Model.
2. Draw the functional block diagram of wide-area networks (WANs), local area networks (LANs) and Wireless LANs (WLANs) describe the function of each block.
3. For a given requirement (small scale) of wide-area networks (WANs), local area networks (LANs) and Wireless LANs (WLANs) design it based on the market available component
4. Understand the functionality of each layer of TCP/IP protocol suite and developed the network programming for a given problem.
5. Configure DNS DDNS, TELNET, EMAIL, File Transfer Protocol (FTP), WWW, HTTP, SNMP, Bluetooth, Firewalls using open source available software and tools.

**Compiler Design LAB**
**PCC-CS691**

| Course Objective |
| --- |
| 1. To implement different phases of compiler |

| |
|---|
| 2. To give exposure to compiler writing tools. |
| **Assignment 1:**<br>Study on GCC compiler |
| **Assignment 2:**<br>Implementation of FA, Tokenization, Use of LEX / FLEX / JFLEX tool to implement lexical analyzer |
| **Assignment 3:**<br>Implementation of top-down and bottom-up parsing algorithm, Use of YACC / BISON / CUP tool to generate parser |
| **Assignment 4:**<br>Implementation of Syntax Directed Definition / Abstract Syntax Tree / Type checking |
| **Assignment 5:**<br>Implementation of symbol table |
| **Assignment 6:**<br>Generation of intermediate code |
| **Assignment 7:**<br>Implementation of code optimization techniques |
| **Course Outcomes** |
| **After completion of the course, students would be able to:** |
| • Use different available tools to implement compiler |
| • Work in the development of compiler for a new computer language in the industry |

**Computer Networks LAB**

**PCC-CS692**

| | | | |
|---|---|---|---|
| **YEAR- 3RD** | | **SEMESTER- VI** | |
| **PCC-CS692** | **Computer Networks LAB** | **0L:0T: 4P** | **2 credits** |
| **Pre-requisites** | **PCC-CS 403** | | |

**Detailed contents:**

**Experiments should include but not limited to :**

1. Familiarization with:
   - Networking cables (CAT5, UTP)
   - Connectors (RJ45, T-connector)
   - NIC Installation & Configuration (Windows/Linux)
   - Hubs, Switches
2. TCP/UDP Socket Programming
3. Multicast & Broadcast Sockets

4. Implementation of a Prototype Multithreaded Server
5. Implementation of
   Data Link Layer Flow Control Mechanism (Stop & Wait, Sliding Window)
   Data Link Layer Error Detection Mechanism (Cyclic Redundancy Check)
   Data Link Layer Error Control Mechanism (Selective Repeat, Go Back N)

**Advanced Algorithms**
**PEC-CS601A**

| | |
|---|---|
| **Course Code** | PEC-CS601A |
| **Course Name** | Advanced Algorithms |
| **Credits** | 3 |
| **Pre-Requisites** | UG level course in Algorithm Design and Analysis |

Total Number of Lectures: 36

| **COURSE OBJECTIVE** |
|---|
| • To equip students with different design paradigms and data structures to solve advanced algorithmic problems. |
| • Students should be able to understand different classes of problems concerning their computational hardness. |
| • The students should be able to choose or design an efficient approximation algorithm to solve a computationally hard problem. |

| **LECTURE WITH BREAKUP** | **NO. OF LECTURES** |
|---|---|
| **Unit1** | 4 |
| **Sorting:** Review of various sorting algorithms, topological sorting<br>**Graph:** Definitions and Elementary Algorithms: Shortest path by BFS, shortest path in edge-weighted case (Dijkasra's), depth-first search and computation of strongly connected components, emphasis on correctness proof of the algorithm<br>and time/space analysis, example of amortized analysis. | |

| | |
|---|---|
| **Unit 2**<br>**Matroids:** Introduction to greedy paradigm, algorithm to compute a maximum weight maximal independent set. Application to MST.<br>**Graph Matching:** Algorithm to compute maximum matching. Characterization of maximum matching by augmenting paths, Edmond's Blossom algorithm to<br>compute augmenting path. | 7 |
| **Unit 3**<br>**Flow-Networks:** Maxflow-mincut theorem, Ford-Fulkerson Method to compute maximum flow, Edmond-Karp maximum-flow algorithm.<br>**Matrix Computations:** Strassen's algorithm and introduction to divide and conquer paradigm, inverse of a triangular matrix, relation between the time<br>complexities of basic matrix operations, LUP-decomposition. | 8 |
| **Unit 4**<br>**Shortest Path in Graphs:** Floyd-Warshall algorithm and introduction to dynamic programming paradigm. More examples of dynamic programming.<br>**Modulo Representation of integers/polynomials:** Chinese Remainder Theorem, base-representation and modulo-representation of numbers. Extension to polynomials. | 6 |
| **Unit 5**<br>**Linear Programming:** Geometry of the feasibility region and Simplex algorithm<br>**NP-completeness:** Examples, proof of NP-hardness and NP-completeness.<br>**One or more of the following topics based on time and interest**<br>Approximation algorithms, Randomized Algorithms, Interior Point Method, Advanced Number Theoretic Algorithm | 6 |
| **Unit 6**<br>Recent Trends in problem solving paradigms using recent searching and sorting<br>techniques by applying recently proposed data structures. | 5 |

| **COURSE OUTCOMES** |
|---|
| After completion of course, students would be able to: |
| • Analyze the complexity/performance of differentalgorithms. |
| • Determine the appropriate data structure for solving a particular set ofproblems. |
| • Categorize the different problems in various classes according to theircomplexity. |

- Students should have an insight of recent activities in the field of the advanced data structure.

**References:**


1. "Introduction to Algorithms" byCormen, Leiserson, Rivest,Stein.
2. "The Design and Analysis of Computer Algorithms" by Aho, Hopcroft,Ullman.
3. "Algorithm Design" by Kleinberg andTardos.


**Distributed Database**
**Subject Code: PEC-CS601B**
**Contacts: 3L Credits: 3**
**Pre-requisites: PCC-CS 502**

**Course Objectives:**

1. To understand the different issues involved in the design and implementation of a distributed database system.
2. To understand distributed query processing and query optimization.
3. To understand distributed transaction management, concurrency control, deadlock management and reliability issues.
4. To develop an understanding of some current topics such as Mobile Databases, Distributed Object Management, Multi-databases

**Detailed contents:**

**Module 1: [5L]**

**Introduction:** Distributed data processing; What is a DDBS; Advantages and disadvantages of DDBS; Overview of database and computer network concepts

**Distributed Database Management System Architecture:** Transparencies in a distributed DBMS; Distributed DBMS architecture.

**Module2: [ 10L]**

**Distributed Database Design:** Reference Architecture; Distributed design issues; Fragmentation; Data allocation; View management; Integrity Control.

**Query Processing Issues:** Objectives of query processing; Layers of query processing; Query decomposition; Localization of distributed data.

**Module3: [ 12L]**

**Distributed Query Optimization:** Factors governing query optimization; Centralized query optimization; Ordering of fragment queries; Distributed query optimization algorithms.

**Transaction Management:** Transaction concept; Goals of transaction management; Characteristics of transactions;

**Concurrency Control:** Concurrency control in DDBSs; Distributed concurrency control algorithms; Deadlock management

**Module4: [ 5L]**

Reliability issues in DDBSs; Types of failures; Reliability techniques; Commit protocols; Recovery protocols.

**Module5:[ 4L]**

**Advanced Topics:** Mobile Databases, Distributed Object Management, Multi-databases

**Text book**:
1. "Principles of Distributed Database Systems", by M.T. Ozsu and PValduriez, Prentice-Hall,
2. "Distributed Databases: Principles and Systems", by Stefano Ceri, Giuseppe Pelagatti; McGraw Hill

**Reference books:**
1. Distributed Database Systems, D. Bell and J. Grimson, Addison- Wesley, 1992.

**Course Outcomes:**
On completion of the course students will be able to
1. Get an in-depth knowledge about distributed database design and its implementation.
2. Explain the purpose of query optimization and the steps involved in generating efficient query plans, and how distributed query evaluation differs than centralized databases.
3. Get an in-depth knowledge about the transaction processing and concurrency control in distributed database.
4. Understand different distributed database recovery protocol and how deadlocks are handled in distributed databases.

**Real Time Systems**
**Course Code: PEC-CS601C**

| | |
|---|---|
| Course Code: PEC-CS601C | Category: Professional Elective Courses |
| **Course Title: Real Time Systems** | Semester: 6th (Elective-II) |
| L-T-P: 3-0-0 | Credit: 3 |
| **Teaching Scheme**<br>Theory: 3hrs/week | **Examination Scheme**<br>Class Tests & Assignments: 25 marks<br>Attendance: 5 marks<br>End Semester Exam: 70 marks |
| **Objective:**<br>**The course is a survey of issues in the design of real time computer systems. It aims to:**<br>• provide a thorough understanding of well known models for reasoning about the performance of real time computer based systems,<br>• enable students to describe and apply commonly used abstract models and terminology for real-time scheduling and resource management,<br>• understand and apply the fundamental theorems of deadline and fixed priority approaches,<br>• evaluate the suitability of an operating system for real-time applications<br>• provide the students sufficient exposure to the properties of at least one freely available real time operating system | |
| **Pre-requisites:** Undergraduate level course on Operating Systems | |

***Detailed Syllabus***

| Unit | Content | Hrs/Unit |
|---|---|---|
| 1 | **Typical Real Time Applications**<br>Examples of digital control such as sampled data systems, example of software controlled structure of (say) a flight controller, example of air traffic/flight control hierarchy. Examples of Real Time Databases, Multimedia Applications, etc.<br><br>**Types of Real Time applications**<br>Purely Cyclic, Mostly Cyclic, Asynchronous and somewhat predictable, Asynchronous and unpredictable.<br><br>**Hard real time systems and Soft real time systems**<br>Definitions: Jobs/Tasks, Processors, Release times, Deadlines, Timing constraints,<br>Hard and Soft timing constraints,<br>Reasons for requiring timing guarantees. | 10 |

| | | |
|---|---|---|
| | **A Reference Model of Real Time Systems**which is characterized by three elements: (1) a workload model that describes the applications supported by the system, (2) a resource model that describes the system resources available to the applications, and (3) algorithms that define how the application system uses the resources at all times.<br>Temporal parameters of real time workload,Periodic Task Model,Precedence constraints and data dependencies among real time tasks/jobs, precedence graph and task graph, other types of dependencies such as temporal dependencies,Functional parameters such as preemptivity of jobs, criticality of jobs, etc.<br>Resource parameters of jobs and parameters of resources,Sheduling hierarchy, feasibility, optimality, and performance measures of schedules, interactions among schedulers. | |
| 2 | **Commonly used approaches to Hard Real-Time Scheduling**<br><br>Basic concepts of Dynamic (where jobs are dispatched dynamically to processors) vs Static systems, Concept of effective release times and deadlines, weighted Round-Robin approach to scheduling tasks in Real-time systems, Challenges in validating timing constraints in priority driven systems, Off-line vs On-line Scheduling, Competitive factor of on-line scheduling algorithm<br><br>*Clock-driven scheduling:*<br><br>Static, timer-driven scheduler, General structure of cyclic schedules using frames and major cycles, Frame size constraints, Job slices, Cyclic executives, Techniques for improving average response time of aperiodic jobs using slack stealing, Basic concepts of scheduling sporadic jobs, acceptance test,<br>Pros and cons of clock driven scheduling<br><br>*Priority-driven scheduling of Periodic Tasks:*<br><br>Fixed priority algorithms (Rate monotonic and Deadline monotonic algorithms), Optimality of the RM and DM algorithms<br><br>Well known Dynamic priority algorithms (Earliest Deadline First, | 11 |

| | | |
|---|---|---|
| | Least Slack Time first algorithms), relevant theorems regarding their optimality; situations when these algorithms are non optimal,<br><br>Concept of Maximum Schedulable Utilization, Schedulable Utilizations of the EDF algorithm (relevant theorems and proof ideas), Schedulability test for the EDF algorithm,<br><br>Schedulability test for fixed-priority tasks with short response times, Time-Demand Analysis | |
| 3 | **Resources and Resource Access Control**<br>Mutual exclusion and critical sections, Resource Conflicts and Blocking, Effects of Resource Contention and Resource Access Control, Non-Preemptive Critical Sections, Basic Priority Inheritance Protocol (definition, properties), Basic Priority Ceiling Protocol (definition and properties), differences between PIP and PCP, Deadlock avoidance by PCP (Priority Ceiling Protocol) | 10 |
| 4 | **Case Study of a (preferably open source) Real time operating system** (eg, FreeRTOS, eCos, etc.) focusing on the features provided by the OS specifically for real time systems | 5 |

***Text book and Reference books***

1. Jane W. S. Liu – Real Time Systems, 2000, Prentice Hall
2. [Relevant online documentation for the real time operating system chosen for case study]

***Course Outcomes***

On completion of the course students will be able to:

1. Identify whether a given system (based on the specifications) is a real time system or not, and also identify the various aspects of the given system according to the reference model of real time systems,
2. Carry out schedulability analysis using deadline and fixed-priority approaches,
3. Describe and apply commonly used abstract models and terminology for real-time scheduling and resource management,
4. Evaluate the suitability of an operating system for real-time applications

**Advanced Computer Architecture**
**PEC-CS601E**

| | |
|---|---|
| Course Code: PEC-CS601E | Category: Professional Elective Courses |
| **Course Title: Advanced Computer Architecture** | Semester: 6th Semester |
| L-T-P: 3-0-0 | Credit: 3 |
| **Teaching Scheme**<br>Theory: 3hrs/week<br>Tutorial: 0 hr/week | **Examination Scheme**<br>Class Tests & Assignments: 25 marks<br>Attendance: 5 marks<br>End Semester Exam: 70 marks |
| **Objective:**<br><br>Throughout this course, students will demonstrate the ability to<br>□ Understand the micro-architectural design of processors.<br>□ Learn about the various techniques used to obtain performance improvement and power savings in current processors. | |
| **Pre-requisites: Basic Computer Architecture syllabus** | |

***Detailed Syllabus***

| Unit | Content | Hrs/Unit (L+T) | Marks/Unit |
|---|---|---|---|
| 1 | **FUNDAMENTALS OF COMPUTER DESIGN:**Computer Architecture and Organization-Review, Review of Fundamentals of CPU, Memory, and IO, Fundamentals of Computer Design, RISC vs. CISC, Trends in technology, power, energy, Cost, Dependability- Performance Evaluation. | 6 | |
| 2 | **INSTRUCTION LEVEL PARALLELISM:** IPL concepts, Concepts of Pipelining, Instruction Pipelining, arithmetic pipelining, Various types of hazards, dynamic pipelining, Exception handling, Pipeline optimization techniques, Compiler techniques for improving performance,dynamic Branch Prediction, | 10 | |

| | | | |
|---|---|---|---|
| | Dynamic Scheduling, Multiple instruction issue, Hardware Based Speculation, Static Scheduling, Multi-threading, Limitations of ILP. | | |
| 3 | **Unit 3: DATA-LEVEL PARALLELISM I:** Vector Processing Principles- Instruction types, Compound, Vector Loops, Chaining, Array Processors (SIMD Extension)- Structure, Parallel Processing Architectures- Taxonomy- SISD, MISD, SIMD, MIMD, Network topologies-Static, Dynamic, Types of Networks, Multiprocessors Architecture- Tightly Coupled and Loosely Coupled Multiprocessors Architecture. Multiprocessors- Multistage Networks, Cache Coherence, Synchronization, Message- passing. | 12 | |
| 4 | **MEMORY:** Hierarchical memory technology: Inclusion, Coherence and locality properties,Types of Storage Devices,Cache memory organizations, Cache Performance, Techniques for reducing cache misses Penalty and Miss Rate, Reducing Hit Time, Virtual memory, Paging. | 8 | |

***Suggested Text books***

**TEXT BOOK:**

- John L Hennessey and David A Patterson, "Computer Architecture A Quantitative Approach", Morgan Kaufmann/ Elsevier, Fifth Edition, 2012.
- Computer Architecture and Parallel Processing- Kai Hwang and A. Brigggs International Edition, McGraw Hill
- Advanced Computer Architecture: D. Sima, T. fountain, P. Kacsuk, Pearson
- Parallel Computer Architecture: D. Culler, J.P.Singh, A.Gupta, Elsevier

**Course Outcome:**

On completion of the course students will be able to:

1. Evaluate performance of different architectures with respect to various parameters.
2. Analyze performance of different IPL techniques.
3. Identify cache and memory related issues in multi-processors.

**Computer Graphics**
**PEC-CS602A**

| | |
|---|---|
| Course Code | PEC-CS602A |
| Course Name | Computer Graphics |
| Credits | 3 |
| L-T-P | 3-0-0 |
| Pre-Requisites | |

**Total Number of Lectures: 36**

| Course Objective |
|---|
| The aim of the course is to<br>i) Introduce graphic system and its components and related algorithms.<br>ii) Make familiar with 2D and 3D computer graphics.<br>iii) Provide an understanding of mapping from world coordinates to device coordinates, clipping and projection.<br>iv) Introduce about modelling, texture, lighting, ray tracing and animation. |

| **LECTURES WITH BREAKUP** | **NO. OF LECTURES** |
|---|---|
| **Unit 1:**<br>Introduction to computer graphics & graphics systems : Overview of computer graphics, representing pictures, preparing, presenting & interacting with pictures for presentations; Visualization & image processing; RGB color model, direct coding, lookup table; storage tube graphics display, Raster scan display, 3D viewing devices, Plotters, printers, digitizers, Light pens etc.; Active & Passive graphics devices; Computer graphics software. | **6** |
| **Unit 2:**<br>Scan conversion : Points & lines, Line drawing algorithms; DDA algorithm, Bresenham's line algorithm, Circle generation algorithm; Ellipse generating algorithm; scan line polygon, fill algorithm, boundary fill algorithm, flood fill algorithm. | **8** |
| **Unit 3:**<br>2D transformation & viewing : Basic transformations: translation, rotation, scaling; Matrix representations & homogeneous coordinates, transformations between coordinate systems; reflection shear; Transformation of points, lines, parallel lines, | **13** |

| | |
|---|---|
| intersecting lines. Viewing pipeline, Window to view port co-ordinate transformation, clipping operations, point clipping, line clipping, clipping circles, polygons & ellipse. Cohen and Sutherland line clipping, Sutherland-Hodgeman Polygon clipping, Cyrus-beck clipping method | |
| **Unit 4:**<br>3D transformation & viewing : 3D transformations: translation, rotation, scaling & other transformations. Rotation about an arbitrary axis in space, reflection through an arbitrary plane; general parallel projection transformation; clipping, view port clipping, 3D viewing. | **5** |
| **Unit 5:**<br>Curves and Surfaces: Curve representation, surfaces, designs, Bezier curves, B-spline curves, end conditions for periodic B-spline curves, rational B-spline curves. Depth comparison, Z-buffer algorithm, Back face detection, BSP tree method, the Painter's algorithm, scan-line algorithm; Hidden line elimination, wire frame methods , fractal - geometry | **6** |
| **Unit 6:**<br>Color & shading models : Light & color model; interpolative shading model; Texture.<br>Introduction to Ray-tracing: Human vision and color, Lighting, Reflection and transmission models.<br>Introduction to Computer Animation | **10** |
| **Course Outcomes** | |
| **After completion of the course, students would be able to:** | |
| ● Explain basics of computer graphics, its components and related algorithms. | |
| ● Explain and design various 2D and 3D computer graphics algorithms. | |
| ● Describe the importance of viewing and projections. | |
| ● Solve graphics programming issues including modelling, texture, lighting, ray tracing and animation | |

**References:**


1. Hearn, Baker – "Computer Graphics (C version 2nd Ed.)" – Pearson education
2. Z. Xiang, R. Plastock – " Schaum's outlines Computer Graphics (2nd Ed.)" – TMH
3. D. F. Rogers, J. A. Adams – "Mathematical Elements for Computer Graphics (2nd Ed.)" – TMH

**Information Theory and Coding**
**PEC-CS602C**

| | |
|---|---|
| **Course Code** | PEC-CS602C |
| **Course Name** | Information Theory and Coding |
| **Credits** | 3 |
| **Pre-Requisites** | Basic Mathematics |

**COURSE OBJECTIVES**

**Total Number of Lectures: 36**

| |
|---|
| • To understand error–control coding. |
| • To understand encoding and decoding of digital data streams. |
| • To be familiar with the methods for the generation of these codes and their decoding techniques. |

| **LECTURE WITH BREAKUP** | **NO.OF LECTURES** |
|---|---|
| ***Source Coding***<br><br>Uncertainty and information, average mutual information and entropy, information measures for continuous random variables, source coding theorem, Huffman codes. | **5** |
| ***Channel Capacity And Coding***<br><br>Channel models, channel capacity, channel coding, information capacity theorem, The Shannon limit. | **5** |
| ***Linear And Block Codes For Error Correction*** | **7** |

| | |
|---|---|
| Matrix description of linear block codes, equivalent codes, parity check matrix, decoding of a linear block code, perfect codes, Hamming codes. | |
| ***Cyclic Codes***<br><br>Polynomials, division algorithm for polynomials, a method for generating cyclic codes, matrix description of cyclic codes, Golay codes. | **6** |
| ***BCH Codes***<br><br>Primitive elements, minimal polynomials, generator polynomials in terms of minimal polynomials, examples of BCH codes. | **7** |
| ***Convolutional Codes***<br><br>Tree codes, trellis codes, polynomial description of convolutional codes, distance notions for convolutional codes, the generating function, matrix representation of convolutional codes, decoding of convolutional codes, distance and performance bounds for convolutional codes, examples of convolutional codes, Turbo codes, Turbo decoding. | **6** |

**COURSE OUTCOMES**

**After completion of course, students would be able to:**

| |
|---|
| 1. Understand the concept of information and entropy. |
| 2. Understand Shannon's theorem for coding. |
| 3. Calculation of channel capacity. |
| 4. Apply coding techniques. |

**References:**


1. N. Abramson, Information and Coding, McGraw Hill, 1963.
2. M. Mansurpur, Introduction to Information Theory, McGraw Hill, 1987.
3. R.B. Ash, Information Theory, Prentice Hall, 1970.
4. Shu Lin and D.J. Costello Jr., Error Control Coding, Prentice Hall, 1983.
5. Ranjan Bose, Information theory, coding and cryptography; TMH.

- OEC-CS601, OPEN ELECTIVE I, Soft Skills and Interpersonal Communication, Credit 3
- **Project, PROJ-CS691, Project, Credit 3**

## Computer Science & Engineering
## Semester VII( 4th Year)

**Code: PEC-CS701A**
**Ad-Hoc and Sensor Networks**
**3L:0T:0P 3 Credits**
**Total Number of Lectures: 36**

| | |
|---|---|
| **Course Code** | **PES-CS701A** |
| **Course Name** | **AD-HOC AND SENSOR NETWORKS** |
| **Credits** | **3L:0T:0P 3 Credits** |
| **Pre-Requisites** | **Computer Networks** |

**Total Number of Lectures: 36**

| **COURSE OBJECTIVES** |
|---|
| **7. To Understand the design issues, characteristics, challenges in Ad-Hoc and Sensor Networks.** |
| **8. To learn the different types routing protocols for Ad-Hoc and Sensor Networks.** |
| **9. To gain knowledge about the Transport Layer and Security Protocols for Ad-Hoc and Sensor Networks.** |
| |

**MODULE 1**

**INTRODUCTION [4L]**

Fundamentals of Wireless Communication Technology – Overview of wireless networks, types, infrastructure-based and infrastructure-less, introduction to MANETs (Mobile Ad-hoc Networks), characteristics -mobile ad hoc networks (MANETs) and wireless sensor networks (WSNs): concepts and architectures, single sensor node and sensor network architecture. Applications of Ad Hoc and Sensor networks. Issues and Challenges in Ad hoc and Sensor Networks.

**MODULE 2**

**Routing Protocols for Ad Hoc Networks [9L]**

Routing Protocols: Design issues, goals and classification. Proactive Vs reactive routing, Hybrid routing protocols, Hierarchical Routing protocols, Power aware routing protocols, Unicast routing algorithms; Multicast routing algorithms – Issues in Designing, Architecture Reference Model, Classifications, Tree-based routing protocols, Mesh-based routing protocols, Quality of Service, Energy Management.

**MODULE 3**

**Transport Layer and Security Protocols for Ad Hoc Networks [6L]**

Issues in designing a Transport layer protocol, Design Goals, Classification of Transport Layer solutions, ad-hoc transport protocols.

Security issues in ad-hoc networks: issues and challenges, network security attacks, Key Management, secure routing protocols.

**MODULE 4 [6L]**

Communication Protocols in WSN: Fundamentals of MAC Protocols, Low duty cycle protocols and wakeup concepts, Contention based protocols -S-MAC, Schedule based protocols-LEACH, IEEE 802.15.4 MAC protocols, Localization and positioning: proximity, trilateration, triangulation, and scene analysis based methods, mathematical basics for the lateration problem, Single hop and Multi-hop localization, Impact of anchor placement.

**MODULE 5 [6L]**

**Routing Protocols for Wireless Sensor Networks**

Gossiping and agent-based unicast forwarding, Energy-efficient unicast, Broadcast

and multicast, Geographic routing, Mobility of nodes : mobile sinks, mobile data collectors, mobile regions. Data centric routing: One-shot interactions (SPIN), Repeated interactions. Data aggregation: Overview, Categories of aggregation operations, placement of aggregation points, When to stop waiting for more data etc.

**MODULE 6**

**Transport Layer and Security Protocols for Wireless Sensor Networks [5L]**

The transport layer and QoS in wireless sensor networks: Quality of service/reliability, Transport protocols. Advanced application support, Security fundamentals, consideration in WSN, Security attacks and threats, attack models etc., Application specific support : important building blocks of sensor network applications, target detection and tracking, contour/edge detection, Field sampling etc.

| **COURSE OUTCOMES** |
|---|
| **After completion of course, students would be able to:** |
| **CO1: Explain the concepts, architectures, issues and applications of Ad-Hoc and Sensor Networks.** |
| **CO2: Analyse and Design the routing protocols for Ad-Hoc and Sensor Networks.** |
| **CO3: Identify and handle security related issuesof Ad-Hoc and Sensor Networks.** |

TEXT BOOK:


1. S R Murthy, B. S. Manoj, Ad Hoc Wireless Networks Architectures and Protocols, Pearson Education.
2. H. Karl, A. Willing, "Protocols and Architectures for Wireless Sensor Networks", John Wiley & Sons, 2007.


REFERENCES:


1. Carlos De Morais Cordeiro, Dharma Prakash Agrawal "Ad Hoc & Sensor Networks: Theory and Applications", World Scientific Publishing Company, 2006.
2. S. K. Sarkar, T G Basavaraju, C Puttamadappa, "Ad Hoc Mobile Wireless Networks: Principles, Protocols, and Applications", (2e), CRC Press, 2016.
3. A. Forster, "Introduction to Wireless Sensor Network", IEEE Press, Willey, 2016.

4. Feng Zhao and Leonides Guibas, “Wireless Sensor Networks”, Elsevier Publication – 2002.
5. R. Jurdak, “Wireless Ad Hoc and Sensor Networks: A Cross-Layer Design Perspective”, Springer Publications, 2007.
6. Kazem Sohraby, Daniel Minoli, & Taieb Znati, “Wireless Sensor Networks-Technology, Protocols, and Applications”, John Wiley, 2007.
7. Anna Hac, “Wireless Sensor Network Designs”, John Wiley, 2003.

**PEC-CS701B**
**Machine Learning**

| | |
|---|---|
| **Course Code** | PEC-CS701B |
| **Course Name** | Machine learning |
| **Credits** | 3 |
| **Pre-Requisites** | Basic Mathematics, Algorithms |

| **COURSE OBJECTIVE** | |
|---|---|
| 1 | To explore supervised and unsupervised learning paradigms of machine learning. |
| 2 | To design and analyse various machine learning algorithms and techniques with a modern outlook focusing on recent advances |
| 3 | To explore Deep learning technique and various feature extraction strategies |

| **LECTURE WITH BREAKUP** | **HOURS** |
|---|---|
| **Unit1:**<br>● Basic concepts of learning, Hypothesis Space, Basic statistics: Probability, Bayes Theorem, Naïve Bayes<br>● Regression Analysis: Correlation, Bivariate and Multivariate regression, Linear, Logistic | **6** |
| Unit2:<br>● Supervised, Unsupervised, Semi-supervised learning, Instance-based learning, k-Nearest Neighbourhood, Ensemble methods – Bagging, Boosting and Stacking<br>● Support Vector Machine: Working principle, Application | 6 |
| **Unit 3:** | **6** |

| | |
|---|---|
| • Decision Trees: Introduction and building, Algorithms used – ID3, Information Gain, Overfitting and Underfitting, Random Forest<br>• Dimensionality reduction: Principle Component Analysis (PCA), Independent Component Analysis (ICA) | |
| **Unit 4:**<br>• Artificial Neural Network: Biological Neuron, MP Neuron, HEBBNet, Perceptron, Multilayer Perceptron<br>• Back-propagation Algorithm, Convolution Neural Network(CNN), Recurrent Neural Network(RNN) | **6** |
| **Unit 5:**<br>• Clustering techniques: k-means, Mean-Shift Clustering, Density-Based Spatial Clustering of Applications with Noise (DBSCAN)<br>• Reinforcement Learning: The Learning Task, Q Learning, Algorithm, Non-deterministic Rewards and Actions | **6** |
| **Unit 6:**<br>• Modeling Sequence/Time-Series Data, Deep Learning and Feature Representation Learning<br>• Evaluating Machine Learning algorithms and Model Selection and real-life applications | **6** |

| |
|---|
| **COURSE OUTCOMES:** After completion of course, students would be able to: |
| • Extract features that can be used for a particular machine learning approach in various real life applications. |
| • To mathematically analyse various machine learning approaches and paradigms. |
| • To compare and contrast pros and cons of various machine learning techniques and to get an insight of when to apply a particular machine learning approach |

**References:**

1. Kevin Murphy, Machine Learning: A Probabilistic Perspective, MIT Press, 2012
2. Trevor Hastie, Robert Tibshirani, Jerome Friedman, The Elements of Statistical Learning,
Springer 2009 (freely available online)
3. Christopher Bishop, Pattern Recognition and Machine Learning, Springer, 2007.

4.Tom M. Mitchell, Machine Learning, McGraw-Hill,1997(freely available online)

**Neural Networks & Deep Learning**
**Code: PEC-CS701C**
**Credit: 3L, Total: 36 Hrs.**

**Description of Course** Artificial Neural Networks are programs that write themselves when given an objective, some training data, and abundant computing power. Recently, these programs have brought about a wide array of future-like innovations, such as self-driving cars, face recognition, and human-like speech generators. This course offers you an introduction to Deep Artificial Neural Networks (i.e. "Deep Learning"). With focus on both theory and practice, we cover models for various applications, how they are trained and tested, and how they can be deployed in real world applications.

**Prerequisites** Basics of probability theory, linear algebra, Introduction to Artificial Intelligence, Computer Architecture/ Design) and calculus at university level.

**Course objectives:**
Objectives: The objective of this course is to cover the fundamentals of neural networks as well as some advanced topics such as recurrent neural networks, long short-term memory cells and convolutional neural networks and Deep Learning. The course also requires students to implement programming assignments related to these topics.

**Module I:**

**Artificial Neural Systems: Preliminaries & Fundamental Concepts and Models of Artificial Neural Systems [3L]**

Classifiers, Approximators, and Autonomous Drivers **,** Simple Memory and Restoration of Patterns, Optimizing Networks, Clustering and Feature Detecting Networks **,** Brief History of Artificial Neural Systems Development Biological Neurons and Their Artificial Models, Models of Artificial Neural Networks, Neural Processing, Learning and Adaptation, Neural Network Learning Rules, Overview of Neural Networks

**Module II:**

**Single-Layer Perceptron Classifiers [3L]**

Classification Model, Features, and Decision Regions, Discriminant Functions,Linear Machine and Minimum Distance Classification,Nonparametric Training Concept, Training and Classification Using the Discrete Perceptron: Algorithm and Example,Single-Layer Continuous Perceptron Networks for LinearlySeparable Classifications, Multi-category Single-Layer Perceptron Networks

**Module III:**

**Multilayer Feedforward Networks [3L]**

Linearly Non-separable Pattern Classification, Delta Learning Rule for Multi-perceptron Layer Generalized Delta Learning Rule, Feedforward Recall and Error Back-Propagation Training, Learning Factors Classifying and Expert Layered Networks, Functional Link Networks

**Module IV:**

**Single-Layer Feedback Networks [3L]**

Basic Concepts of Dynamical Systems, Mathematical Foundations of Discrete-Time Hopfield Networks, Mathematical Foundations of Gradient-Type Hopfield Networks, Transient Response of Continuous-Time Networks,
Relaxation Modeling in Single-Layer Feedback Networks, Example Solutions of Optimization Problems, Summing Network with Digital Outputs, Minimization of the Traveling Salesman Tour Length

**Module V:**

**Matching and Self-Organizing Networks [3L]**

Hamming Net and MAXNET, Unsupervised Learning of Clusters, Counter-propagation Network, Feature Mapping, Self-organizing Feature Maps, Cluster Discovery Network (ART1)

**Module VI:**

**Deep Neural Networks [7L]**

Difficulty of training deep neural networks, Greedy layer wise training, Newer optimization methods for neural networks (Adagrad, adadelta, rmsprop, adam, NAG), second order methods for training, Saddle point problem in neural networks, Regularization methods (dropout, drop connect, batch normalization).

**Module VII [7L]:**

**Recurrent Neural Networks:** Back propagation through time, Long Short Term Memory, Gated Recurrent Units, Bidirectional LSTMs, Bidirectional RNNs,

**Convolutional Neural Networks:** LeNet, AlexNet.

**Module VIII: Generative models [3L]**

Restrictive Boltzmann Machines (RBMs), Introduction to MCMC and Gibbs Sampling, gradient computations in RBMs, Deep Boltzmann Machines.

**Module IX: Recent trends and Applications [4L]**

i) Linear Programming Modeling Network, Character Recognition Networks , Neural Networks Control Applications, Connectionist Expert Systems for Medical Diagnosis , Self-organizing Semantic Maps , Application in Robotics,
ii) Variational Autoencoders, Generative Adversarial Networks, Multi-task Deep Learning, Multi-view Deep Learning, Vision, NLP, Speech

Text Books:


1. Jacek M. Zurada, Introduction to Artificial Neural Systems, JAICO Publishing House, 2006.
2. Deep Learning, Ian Goodfellow and Yoshua Bengio and Aaron Courville, MIT Press, 2016.


References:


1. Simon Haykin, Neural Networks: A Comprehensive Foundation, Macmillan College Publishing Company, 1994.
2. Mohamad H. Hassoun, Foundamentals of Artificial Neural Networks, The MIT Press, 1995.
3. Laurene Fausett, Fundamentals of Neural Networks: Architectures, Algorithms, and Applications, Prentice Hall International, Inc., 1994.
4. B. D. Ripley, Pattern Recognition and Neural Networks, Cambridge University Press., 1996.
5. Neural Networks: A Systematic Introduction, Raúl Rojas, 1996
6. Pattern Recognition and Machine Learning, Christopher Bishop, 2007


Course Outcomes:
Upon completion of the course, the student will be able to

- Understand what Artificial Neural Network is
- Comprehend the concepts of feed forward neural networks
- Analyze the various feedback networks.
- Apply the knowledge of Artificial Neural Network and Deep learning systems to control the real time systems

**Speech and Natural Language Processing**

**PEC-CS702A**

| | |
|---|---|
| **Subject Code:** PEC-CS702A | **Category:** Professional Elective courses |
| **Course Title:** Speech and Natural Language Processing | **Semester:** VII |
| 3L : 0T: 0P | **Credits:** 3 |
| **Pre-requisites:** PCC-CS503, PCC-CS601 | |

[L= Lecture, T = Tutorials, P = Practicals]

**Objectives of the course:**

This course will expose students to the following

1. Provide an in-depth concept of the computational properties of natural languages and algorithms for processing linguistic information.
2. Understand the concepts of morphology, syntax, semantics and pragmatics of the language.
3. Understand statistical language models, POS tagging, machine learning techniques, context free grammar for English language, and text classification.
4. Introduce to work in information retrieval, extraction and analysis using natural language resources.

**Detailed contents:**

| Module | Content | No. of Lecture |
|---|---|---|
| 1 | Regular Expressions and Automata, Introduction to NLP, Regular Expression, Finite State Automata.<br><br>**Tokenization:** Word Tokenization, Normalization, Sentence Segmentation, Named Entity Recognition, Multi Word Extraction, Spell Checking – Bayesian Approach, Minimum Edit Distance.<br><br>**Morphology:** Morphology – Inflectional and Derivational Morphology, Finite State Morphological Parsing, The Lexicon and Morphotactics, Morphological Parsing with Finite State Transducers, Orthographic Rules and Finite State Transducers, | 10 |

| | | |
|---|---|---|
| | Porter Stemmer | |
| 2 | **Language Modeling:** Introduction to N-grams, Chain Rule, Smoothing – Add-One Smoothing, Witten-Bell Discounting; Backoff, Deleted Interpolation, N-grams for Spelling and Word Prediction, Evaluation of language models.<br><br>**Hidden Markov Models and POS Tagging:** Markov Chain, Hidden Markov Models, Forward Algorithm, Viterbi Algorithm, Part of Speech Tagging – Rule based and Machine Learning based approaches, Evaluation | 10 |
| 3 | **Text Classification:** Text Classification, Naïve Bayes' Text Classification, Evaluation, Sentiment Analysis – Opinion Mining and Emotion Analysis, Resources and Techniques<br><br>**Context Free Grammar:** Context Free Grammar and Constituency, Some common CFG phenomena for English, Top-Down and Bottom-up parsing, Probabilistic Context Free Grammar, Dependency Parsing | 8 |
| 4 | **Computational Lexical Semantics:** Introduction to Lexical Semantics – Homonymy, Polysemy, Synonymy, Thesaurus – WordNet, Computational Lexical Semantics – Thesaurus based and Distributional Word Similarity<br><br>**Information Retrieval:** Boolean Retrieval, Term-document incidence, The Inverted Index, Query Optimization, Phrase Queries, Ranked Retrieval – Term Frequency – Inverse Document Frequency based ranking, Zone Indexing, Query term proximity, Cosine ranking, Combining different features for ranking, Search Engine Evaluation, Relevance Feedback | 8 |

**Suggested text books:**


5. "Speech and Language Processing", Jurafsky and Martin, Pearson Education.
6. "Foundation of Statistical Natural Language Processing", Manning and Schutze, MIT Press.


**Course outcomes:**

1. Understand the basic mathematical models and algorithms in the area of NLP.

2. Apply the language resource annotation and machine learning approaches to analyze data and acquired intended information.
3. Understand the concept of text classification and context free grammar.
4. Understand the concept of semantics, information retrieval, extraction and analyzing of intended data from various natural language resources.

**Human Computer Interaction Syllabus (HCI)**
**PEC-CS702B**
**Credits: 3**
**TOTAL: 36 PERIODS**
**OBJECTIVES:** The student should be made to:

- Learn the foundations of Human Computer Interaction
- Be familiar with the design technologies for individuals and persons with disabilities
- Be aware of mobile HCI
- Learn the guidelines for user interface.

**UNIT I: FOUNDATIONS OF HCI** [7 hours]
The Human: I/O channels – Memory – Reasoning and problem solving; The computer: Devices – Memory – processing and networks; Interaction: Models – frameworks – Ergonomics – styles – elements – interactivity- Paradigms.

**UNIT II: DESIGN & SOFTWARE PROCESS** [8 hours]
Interactive Design basics – process – scenarios – navigation – screen design – Iteration and prototyping. HCI in software process – software life cycle – usability engineering – Prototyping in practice – design rationale. Design rules – principles, standards, guidelines, rules. Evaluation Techniques – Universal Design.

**UNIT III: MODELS AND THEORIES** [9 hours]
Cognitive models –Socio-Organizational issues and stake holder requirements – Communication and collaboration models-Hypertext, Multimedia and WWW.

**UNIT IV: MOBILE HCI Mobile** Ecosystem: Platforms, Application frameworks- Types of Mobile Applications: Widgets, Applications, Games- Mobile Information Architecture, Mobile 2.0, Mobile Design: Elements of Mobile Design, Tools. [7 hours]

**UNIT V : WEB INTERFACE DESIGN** Designing Web Interfaces – Drag & Drop, Direct Selection, Contextual Tools, Overlays, Inlays and Virtual Pages, Process Flow. Case Studies. [7 hours]

**OUTCOMES: Upon completion of the course, the student should be able to:**

- Design effective dialog for HCI.
- Design effective HCI for individuals and persons with disabilities.
- Assess the importance of user feedback.
- Explain the HCI implications for designing multimedia/ ecommerce/ e-learning Web sites.
- Develop meaningful user interface.

**TEXT BOOKS:**

1. Alan Dix, Janet Finlay, Gregory Abowd, Russell Beale, "Human Computer Interaction", 3rd Edition, Pearson Education, 2004 (UNIT I, II & III).
2. Brian Fling, "Mobile Design and Development", First Edition , O"Reilly Media Inc., 2009 (UNIT –IV).
3. Bill Scott and Theresa Neil, "Designing Web Interfaces", First Edition, O"Reilly, 2009. (UNIT-V).

**VLSI Design**
**PEC-CS702C**
**Total Number of Lectures: 3**

| | |
|---|---|
| Course Code | PEC-CS702C |
| Course Name | VLSI Design |
| Credits | 3 |
| L-T-P | 3-0-0 |
| Pre-Requisites | Digital Logic |

| Course Objective |
|---|
| i) To provide rigorous foundation in MOS and CMOS digital circuits.<br>ii) CMOS circuit characteristics and their performance.<br>iii) To learn concepts of VLSI Design flow<br>iv) To learn CMOS based static circuits and dynamic circuits,<br>v) To learn how to design static and dynamic circuits by using different CMOS logic families.<br>vi) Give exposure to different steps involved in the fabrication of ICs using |

MOS transistor, CMOS/BICMOS transistors, and passive components.

vii) Explain electrical properties of MOS and BiCMOS devices to analyze the behaviour of inverters designed with various loads.

viii) Provide concept to design different types of logic gates using CMOS inverter and analyze their transfer characteristics.

ix) Provide design concepts to design building blocks of data path of any system using gates.

| **LECTURES WITH BREAKUP** | **NO. OF LECTURES** |
|---|---|
| **Unit 1:** Introduction to IC Technology – MOS, PMOS, NMOS, CMOS & BiCMOS, A brief history, Introduction to VLSI Design, Different types of VLSI design styles: Full custom, standard cell based, gate array based, programmable logic, field programmable gate arrays etc. | **6** |
| **Unit 2:** VLSI Design flow, CMOS logic: PMOS, NMOS and CMOS, Electrical characteristics, operation of MOS transistors as a switch and an amplifier, MOS inverter. | 6 |
| **Unit 3:** Different types of MOS circuits: Dynamic logic, BiCMOS, pass transistors etc. CMOS process, Combinational logic cells, Sequential logic cells, Datapath logic cells, I/O cells. Logical effort, gate array, standard cell and datapath cell design. | **10** |
| **Unit 4:** Introduction to hardware description language (HDL) Verilog/VHDL, VHDL description of combinational circuits, A logic synthesis example. | **8** |
| **Unit 5:** Floor-planning and Placement: I/O and power planning, clock planning. Routing: global and detailed. Example design technique: mapping of architecture to silicon. | **6** |

**Course Outcomes**

**After completion of the course, students would be able to:**

- Implement the logic circuits using MOS and CMOS technology.
- Understand the rapid advances in CMOS Technology

| |
|---|
| • Understand the basics of VHDL.<br>• Develop VHDL models of combinational circuits . |
| • Understand the rapid advances in CMOS Technology<br>• Acquire qualitative knowledge about the fabrication process of integrated circuit using MOS transistors.<br>• Design different types of logic gates using CMOS inverter and analyze their transfer characteristics<br>• Provide design concepts required to design building blocks of data path using gates. |

**References:**


1. Sung-Mo Kang, CMOS Digital Integrated Circuits, 3rd Edition, McGraw-Hill, 2003.
2. Charles Roth, Digital Systems Design using VHDL, 2/e, Cengage Learning, 2012.
3. D.A. Pucknell and K. Eshraghian, *Basic VLSI Design*, PHI Learning Private Limited, 2013.
4. J. Bhasker, *Verilog HDL Synthesis: A Practical Primer*, B.S. Publications, 1998.
5. *Online MOOC Course on Verilog*,http://onlinecourses.nptel.ac.in/noc18_cs48/preview


**Data Analytics**
**PEC-CS702D**
**Contact: 3L Credits: 3**
**Total Number of Lectures: 36**

| | |
|---|---|
| **Course Code** | **Data Analytics PEC-CS702D** |
| **Course Name** | **Data Analytics** |
| **Credits** | **3** |

| Pre-Requisites | **Algorithms and Data Structures:** Understand the concepts of data structures and algorithms. How algorithms work and how and why are they dependent on Data Structures.<br><br>**Database** : Basic knowledge of DBMS<br>Should have knowledge of one Programming Language (Java preferably), Practice of SQL (queries and sub queries), exposure to Linux Environment**.** |
|---|---|

**Total Number of Lectures: 36**

| **COURSE OBJECTIVES** |
|---|
| 1. To explore the fundamental concepts of data analytics. |
| 2. To learn to analyze the big data using intelligent techniques. |
| 3. To understand the applications using Map Reduce Concepts. |
| 4. Understand HDFS Concepts and Interfacing with HDFS |

**Syllabus:**

**MODULE 1:**

**Introduction to Big Data :**
Introduction to Big Data Platform – Challenges of Conventional Systems; Big Data Overview-Data Structures, Analyst Perspective on Data Repositories; State of the Practice in Analytics- BI Versus Data Science, Current Analytical Architecture, Drivers of Big Data, Emerging Big Data Ecosystem and a New Approach to Analytics; Key Roles for the New Big Data Ecosystem; Examples of Big Data Analytics

**MODULE 2:**

**Data Analytics**
Data Analytics – Overview, Data Analytics Lifecycle Overview -Key Roles for a Successful Analytics Project, Background and Overview of Data Analytics Lifecycle - Discovery, Data Preparation, Model Planning,Model Building, Communicate Results,Operationalize; Methodology, Core Deliverables, Key Stakeholders, Data Analyst, Data Scientist

**MODULE 3:**

**Data Analytics Project**

Problem Definition, Data Collection, Cleansing data, Summarizing, Data Exploration, Data Visualization

**MODULE 4:**

**Data Analytics Methods**

Introduction to R, Introduction to SQL, Charts & Graphs- Univariate Graphical Methods - Box-Plots, Histograms, Multivariate Graphical Methods, Data Tools, , Statistical Methods

**MODULE 5:**

**Map Reduce**

Anatomy of a Map Reduce Job Run, Failures, Job Scheduling, Shuffle and Sort, Task Execution, Map Reduce Types and Formats, Map Reduce Features.

**MODULE 6:**

**Hadoop**

Introduction; History of Hadoop; Hadoop - Environment Setup, HDFS Overview, HDFS Operations, Hadoop – Streaming, Hadoop - Multi-Node Cluster, Hadoop Echo System

| **LECTURE WITH BREAKUP** | **NO. OF LECTURES** |
|---|---|
| **MODULE 1:**<br><br>Introduction to Big Data:<br>Introduction to Big Data Platform – Challenges of Conventional Systems; Big Data Overview-Data Structures, Analyst Perspective on Data Repositories; State of the Practice in Analytics- BI Versus Data Science, Current Analytical Architecture, Drivers of Big Data, Emerging Big Data Ecosystem and a New Approach to Analytics; Key Roles for the New Big Data Ecosystem; Examples of Big Data Analytics | [6L] |

| | |
|---|---|
| **MODULE 2:**<br><br>Data Analytics<br>Data Analytics – Overview, Data Analytics Lifecycle Overview - Key Roles for a Successful Analytics Project, Background and Overview of Data Analytics Lifecycle - Discovery, Data Preparation, Model Planning, Model Building, Communicate Results, Operationalize; Methodology, Core Deliverables, Key Stakeholders, Data Analyst, Data Scientist | [6L] |
| **MODULE 3:**<br>Data Analytics Project<br>Problem Definition, Data Collection, Cleansing data, Summarizing, Data Exploration, Data Visualization | [6L] |
| **MODULE 4:**<br><br>Data Analytics Methods<br>Introduction to R, Introduction to SQL, Charts & Graphs- Univariate Graphical Methods - Box-Plots, Histograms, Multivariate Graphical Methods, Data Tools, , Statistical Methods | [6L] |
| **MODULE 5:**<br><br>Map Reduce<br>Anatomy of a Map Reduce Job Run, Failures, Job Scheduling, Shuffle and Sort, Task Execution, Map Reduce Types and Formats, Map Reduce Features. | [6L] |
| **MODULE 6:**<br><br>Hadoop<br><br>Introduction; History of Hadoop; Hadoop - Environment Setup, HDFS Overview, HDFS Operations, Hadoop – Streaming, Hadoop - Multi-Node Cluster, Hadoop Echo System | [6L] |

| COURSE OUTCOME |
|---|
| Upon successful completion of the course students should be able to: |
| CO1: Identify Big Data and its Business Implications. |
| CO2: Develop the ability to build and assess data-based models. |
| CO3: Apply data analytics concepts and methods to solve problems in real-world contexts and will communicate these solutions effectively. |

**Textbooks/References:**


1. Michael Berthold, David J. Hand, "Intelligent Data Analysis", Springer, 2007.
2. Tom White "Hadoop: The Definitive Guide" Third Edition, O'reilly Media, 2012.
3. Chris Eaton, Dirk DeRoos, Tom Deutsch, George Lapis, Paul Zikopoulos, "Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data", McGrawHill Publishing, 2012.
4. Anand Rajaraman and Jeffrey David Ullman, "Mining of Massive Datasets", CUP, 2012.
5. Bill Franks, "Taming the Big Data Tidal Wave: Finding Opportunities in Huge Data Streams with Advanced Analytics", John Wiley& sons, 2012.
6. Glenn J. Myatt, "Making Sense of Data", John Wiley & Sons, 2007.
7. Pete Warden, "Big Data Glossary", O'Reilly, 2011.
8. Jiawei Han, Micheline Kamber "Data Mining Concepts and Techniques", 2nd Edition, Elsevier, Reprinted 2008.
9. Da Ruan, Guoquing Chen, Etienne E.Kerre, Geert Wets, "Intelligent Data Mining", Springer, 2007.
10.Paul Zikopoulos, Dirkde Roos, Krishnan Parasuraman, Thomas Deutsch, James Giles , David Corrigan, "Harness the Power of Big Data The IBM Big Data Platform", Tata McGraw Hill Publications, 2012.
11. Arshdeep Bahga, Vijay Madisetti, "Big Data Science & Analytics: A Hands On Approach ",VPT, 2016
12. Bart Baesens "Analytics in a Big Data World: The Essential Guide to Data Science and its Applications (WILEY Big Data Series)", John Wiley &Sons,2014
13. Seema Acharya, Subhasini Chellappan, "Big Data Analytics" Wiley 2015.


**Indian Music System**

**OEC-CS701B**
**Contacts: 36 L**
**Credits: 3**

**Objective:**

1. Learning Indian Music system
2. Learning Indian Music Instruments
3. Computational analysis of Music pattern
4. Recommendation system design.

**Part I :Indian classical Music classification**, music three main elements - melody, rhythm and harmony,The Language of Music, Notes. Hindustani (North Indian) classical music, an octave into 12 notes. Carnatic classical Music.Natural Origin of Notes.

Part II: Defining Ragas, Elements of a Raga, Aroh & avroh,Ornamentation,Focal points – vadi & samvadi,Deergha (elongated) and alpa (weak) notes,Nyasa (landing notes),Raga Classification Systems,Raga Classification by Structure (jaati), Raga Classification by Scale (thaat), Raga Classification by Family (raagang), Difficult Ragas,Rhythm (taal) in Indian Classical Music,Ornamentation in Indian Classical Music (alankar),Notating Indian Classical Music, Fixed Raga Compositions (*bandish*),Improvisation in Indian Classical Music.

Part III: Indian Music Instruments, Tabla, Guiter,Sitar,Sitar.Khartal. Manjira. Mridanga. Saraswati Veena. Sarod. Shehnai, Ghatam. Pluck string and stuck string.musical instrument segregation system

Part IV: Computational Music Analysis:Topological Structures in Computer-Aided Music Analysis, Computational Analysis of Musical Form, Contextual Set-Class Analysis,Chord- and Note-Based Approaches to Voice Separation,Analyzing Symbolic Music with Probabilistic Grammars,Interactive Melodic Analysis,Wavelet-Based Approach to Pattern Discovery in Melodies,Composer Classification Models for Music-Theory Building, Pattern Mining in Folk Music Analysis. Object Orinted Modelling of Music.Leveraging Machine Learning and Soft Computing Techniques to Investigate Raag Formation in Indian Classical Music, Internet of Music Things.

Part V: Music Recommendation system: Filtering methods, Context awareness by case-based reasoning in a music recommendation system, Emotion based music recommendation system, Customer Sentiment Analysis for recommendation. Healthcare system recommendation of music.

Reference: 1. Music Theory: from Absolute Beginner to Expert, Nicolas Carter

2. Indian Classical Music Pt. Ravi Shankar Prasanna

3. Computational Music Analysis, Editors: Meredith, David (Ed.)

- **BASIC SCIENCE COURSE, Biology – BS-B701, Credit 3**
- **Project, PROJ-CS791, Project II, Practical, 12 Hrs, Credit 6**

## Computer Science & Engineering
## SEMESTER-VIII(4th Year)

**Quantum Computing**
**PEC-CS801B**

| | |
|---|---|
| Course Code | PEC-CS801B |
| Course Name | Quantum Computing |
| Credits | 3 |
| L-T-P | 3-0-0 |
| Pre-Requisites | Digital Logic |

**Total Number of Lectures: 36**

| |
|---|
| Course Objective |

i) i)The objective of this course is to provide the students an introduction to quantum computation. Much of the background material related to the algebra of complex vector spaces and quantum Mechanics is covered within the course.
ii) ii)The objective of this course is to impart necessary knowledge to the learner so that he/she can develop and implement algorithm and write programs using these algorithm.

| **LECTURES WITH BREAKUP** | **NO. OF LECTURES** |
|---|---|
| **Unit 1:**<br>**Unit 1:**<br>1. Introduction to Quantum Computing (6 Hours)<br>1.1 Motivation for studying Quantum Computing<br>1.2 Major players in the industry (IBM, Microsoft, Rigetti, D-Wave etc.)<br>1.3 Origin of Quantum Computing<br>1.4 Overview of major concepts in Quantum Computing<br>· Qubits and multi-qubits states, Bra-ket notation.<br>· Bloch Sphere representation<br>· Quantum Superposition<br>· Quantum Entanglement | **6** |
| **Unit 2:**<br>2. Math Foundation for Quantum Computing<br>2.1 Matrix Algebra: basis vectors and orthogonally, inner product andHilbert spaces, matrices and tensors, unitary operators and projectors, Dirac notation, Eigen values and Eigen vectors. | 6 |

| | |
|---|---|
| **Unit 3:**<br>3. Building Blocks for Quantum Program<br>3.1 Architecture of a Quantum Computing platform<br>3.2 Details of q-bit system of information representation:<br>· Block Sphere<br>· Multi-qubits States<br>· Quantum superposition of qubits (valid and invalid superposition)<br>· Quantum Entanglement<br>· Useful states from quantum algorithmic perceptive e.g. Bell State<br>· Operation on qubits: Measuring and transforming using gates.<br>· Quantum Logic gates and Circuit: Pauli, Hadamard, phase shift,<br>Controlled gates, Deutsch, swap etc.<br>3.3 Programming model for a Quantum Computing Program<br>· Steps performed on classical computer<br>· Steps performed on Quantum Computer<br>· Moving data between bits and qubits. | **10** |
| **Unit 4: Unit 4: Quantum Algorithms**<br><br>**4.1 Basic techniques exploited by quantum algorithms.**<br>· Amplitude amplification<br>· Quantum Fourier Transform<br>· Phase Kick-back<br>· Quantum Phase estimation<br>· Quantum Walks<br>**4.2 Major Algorithms**<br>· Shor's Algorithm<br>· Grover's Algorithm<br>· Deutsch's Algorithm<br>· Deutsch -Jozsa Algorithm | **8** |
| **Unit 5:**<br><br>**Noise and error correction:** Graph states and codes, Quantum error correction,<br>fault-tolerant computation. | **6** |
| **Course Outcomes** | |
| **After completion of the course, students would be able to:** | |
| ● Explain the working of a Quantum Computing program, its architecture and | |

programmodel

- Develop quantum logic gate circuits
- Develop quantum algorithm
- Program quantum algorithm on major toolkits

**References:**


1. Sung-Mo Kang, CMOS Digital Integrated Circuits, 3rd Edition, McGraw-Hill, 2003.
2. Charles Roth, Digital Systems Design using VHDL, 2/e, Cengage Learning, 2012.
3. D.A. Pucknell and K. Eshraghian, *Basic VLSI Design*, PHI Learning Private Limited, 2013.
4. J. Bhasker, *Verilog HDL Synthesis: A Practical Primer*, B.S. Publications, 1998.
5. Online MOOC Course on Verilog,http://onlinecourses.nptel.ac.in/noc18_cs48/preview


**Cryptography & Network Security**
**PEC-CS801C**
**Contracts: 3L, Credits- 3**
**Total: - 36 Lectures**

**Course description and objectives:**
This Course focuses towards the introduction of network security using various cryptographic algorithms. Underlying network security applications. It also focuses on the practical applications that have been implemented and are in use to provide email and web security.

**Module I:**
**Introduction [4L]**
Need for Network Security, Security approaches, Principles of Security, Types of attack.
**Module II:**
**Cryptography: Fundamentals [6L]**
Introduction, Plaintext & Cipher text, Substitution Techniques, Transposition Techniques, Encryption & Decryption,
Symmetric & Asymmetric key Cryptography, Key Range & Key Size.
**Module III:**
**Symmetric Key Algorithm [8L]**

Algorithm types and Modes, Overview of Symmetric Key Cryptography, DES (Data Encryption Standard) algorithm, IDEA (International Data Encryption Algorithm) algorithm, RC5 (Rivest Cipher 5) algorithm.

**Module IV:**

**Asymmetric Key Algorithms [8L]**

Overview of Asymmetric key Cryptography, RSA algorithm, Symmetric and Asymmetric key Cryptography Comparison, Digital Signature, Basic concepts of Message Digest and Hash Function , MD5, SHA-1.

**Module V:**

**Authentication Techniques, Internet Security Protocols [4L]**

Basic Concepts, SSL protocol, Authentication Basics, Password, Authentication Token, Certificate based Authentication, Biometric Authentication.

**Module VI:**

**Electronic Mail Security [3L]**

Basics of mail security, Pretty Good Privacy, S/MIME.

**Module VII:**

**Firewall [3L]**

Introduction, Types of firewall, Firewall Configurations, DMZ Network

Text:

1. "Cryptography and Network Security", William Stallings, 2nd Edition, Pearson Education Asia
2. "Network Security private communication in a public world", C. Kaufman, R. Perlman and M. Speciner,
Pearson
3. Cryptography & Network Security: Atul Kahate, TMH.

Reference :

1. "Network Security Essentials: Applications and Standards" by William Stallings, Pearson
2. "Designing Network Security", Merike Kaeo, 2nd Edition, Pearson Books
3. "Building Internet Firewalls", Elizabeth D. Zwicky, Simon Cooper, D. Brent Chapman, 2nd Edition, Oreilly
4. "Practical Unix & Internet Security", Simson Garfinkel, Gene Spafford, Alan Schwartz, 3rd Edition, Oreilly

**Course Outcomes:**

· Understand the most common type of cryptographic algorithm
· Understand the Public-Key Infrastructure
· Understand security protocols for protecting data on networks
· Be able to digitally sign emails and files
· Understand vulnerability assessments and the weakness of using passwords for authentication
· Be able to perform simple vulnerablility assessments and password audits
· Be able to configure simple firewall architectures
· Understand Virtual Private Networks

**PEC-CS801D**

**Cloud Computing**
**Contact: 3L Credits: 3**
**Total Number of Lectures: 36**

| | |
|---|---|
| **Course Code** | **Program Elective PEC-CS801D** |
| **Course Name** | **Cloud Computing** |
| **Credits** | **3** |
| **Pre-Requisites** | **Operating System, Distributed and Parallel Computing systems, Computer Network** |

**Total Number of Lectures: 36**

| **COURSE OBJECTIVES** |
|---|
| Students should be able to understand the evolution of Cloud Computing from the existing technologies. |
| To develop conceptual understanding of Cloud Computing and have knowledge on the various issues in Cloud Computing. |
| To be familiar with the emerging technologies as the next generation computing paradigms. |
| To understand how a real world problem can be mapped to the Cloud Computing domain and to solve different industry level problems. |

**Syllabus:**

**Module1**

**INTRODUCTION**

Introduction to Cloud Computing - Overview of Computing Paradigm; Recent trends in Computing - Grid Computing, Cluster Computing, Distributed Computing, Utility Computing, Cloud Computing; Business driver for adopting cloud computing; ; Underlying Principles of Parallel and Distributed Computing; Parallel and Distributed Systems; Distributed Computing System Models – Minicomputer Model, Workstation Model, Workstation-Server Model, Processor-Pool Model, Hybrid Model; Network Operating Systems and Distributed Operating Systems; Conventional Computing vs. Cloud Computing;

**Module 2**

**CLOUD ARCHITECTURE AND SERVICES**

Introduction to Cloud Computing - Cloud Computing (NIST Model); History of Cloud Computing, Cloud service providers; Properties, Characteristics & Disadvantages - Pros and Cons of Cloud Computing, Benefits of Cloud Computing, Cloud computing vs. Cluster computing vs. Grid computing

Cloud Computing Architecture - Cloud computing stack- Comparison with traditional computing architecture (client/server), Services provided at various levels, How Cloud Computing Works, Role of Networks in Cloud computing, protocols used, Role of Web services; Service Models (XaaS) -Infrastructure as a Service(IaaS), Platform as a Service(PaaS), Software as a Service(SaaS); Deployment Models - Public cloud, Private cloud, Hybrid cloud, Community cloud.

**Module 3**

**INTRODUCTION TO VIRTUALIZATION**

Introduction to Virtualization: Virtualization and cloud computing - Need of virtualization – cost, administration, fast deployment, reduce infrastructure cost – limitations Types of hardware virtualization: Full virtualization - partial virtualization - para virtualization Desktop virtualization: Software virtualization – Memory virtualization - Storage virtualization – Data virtualization – Network virtualization

Hypervisors and Virtual machines: Server Virtualization: Understanding Server Virtualization, types of server virtualization, Virtual machine basics, types of virtual machines, hypervisor concepts and types.

**Module 4**

**CLOUD SECURITY**

Infrastructure Security - Network level security, Host level security, Application level security; Data security and Storage - Data privacy and security Issues, Jurisdictional issues raised by Data location; Identity & Access Management; Access Control; Trust, Reputation, Risk; Authentication in cloud computing, Client access in cloud, Cloud contracting Model, Commercial and business considerations.

**Module5**

**CLOUD TECHNOLOGIES AND ADVANCEMENTS**

Hadoop; MapReduce ; Virtual Box ; Google App Engine ; Programming

Environment for Google App Engine ; Open Stack ;

**Module 6**

**ADVANCES IN CLOUD COMPUTING**

Mobile Cloud Computing(MCC), Offloading in MCC, Load balancing, Crowdsensing, Trust management in MCC; Introduction to Fog, Edge and Dew Computing, Algorithm and applications of Fog, Edge and Dew computing.

| **LECTURE WITH BREAKUP** | **NO. OF LECTURES** |
|---|---|
| **INTRODUCTION**<br><br>Introduction to Cloud Computing - Overview of Computing Paradigm; Recent trends in Computing - Grid Computing, Cluster Computing, Distributed Computing, Utility Computing, Cloud Computing; Business driver for adopting cloud computing; ; Underlying Principles of Parallel and Distributed Computing; Parallel and Distributed Systems; Distributed Computing System Models – Minicomputer Model, Workstation Model, Workstation-Server Model, Processor-Pool Model, Hybrid Model; Network Operating Systems and Distributed Operating Systems; Conventional Computing vs. Cloud Computing; | **[6L]** |
| Module 2<br><br>CLOUD ARCHITECTURE AND SERVICES<br><br>Introduction to Cloud Computing - Cloud Computing (NIST Model); History of Cloud Computing, Cloud service providers; Properties, Characteristics & Disadvantages - Pros and Cons of Cloud Computing, Benefits of Cloud Computing, Cloud computing vs. Cluster computing vs. Grid computing<br><br>Cloud Computing Architecture - Cloud computing stack- Comparison with traditional computing architecture (client/server), | **[6L]** |

| | |
|---|---|
| Services provided at various levels, How Cloud Computing Works, Role of Networks in Cloud computing, protocols used, Role of Web services; Service Models (XaaS) -Infrastructure as a Service(IaaS), Platform as a Service(PaaS), Software as a Service(SaaS); Deployment Models - Public cloud, Private cloud, Hybrid cloud, Community cloud. | |
| Module 3<br><br>INTRODUCTION TO VIRTUALIZATION<br><br>Introduction to Virtualization: Virtualization and cloud computing - Need of virtualization – cost, administration, fast deployment, reduce infrastructure cost – limitations Types of hardware virtualization: Full virtualization - partial virtualization - para virtualization Desktop virtualization: Software virtualization – Memory virtualization - Storage virtualization – Data virtualization – Network virtualization<br><br>Hypervisors and Virtual machines: Server Virtualization: Understanding Server Virtualization, types of server virtualization, Virtual machine basics, types of virtual machines, hypervisor concepts and types. | **[6L]** |
| Module 4<br><br>CLOUD SECURITY<br><br>Infrastructure Security - Network level security, Host level security, Application level security; Data security and Storage - Data privacy and security Issues, Jurisdictional issues raised by Data location; Identity & Access Management; Access Control; Trust, Reputation, Risk; Authentication in cloud computing, Client access in cloud, Cloud contracting Model, Commercial and business considerations. | **[6L]** |
| Module5<br><br>CLOUD TECHNOLOGIES AND ADVANCEMENTS<br><br>Hadoop; MapReduce ; Virtual Box ; Google App Engine ; Programming Environment for Google App Engine ; Open Stack ; | **[6L]** |
| **ADVANCES IN CLOUD COMPUTING** | **[6L]** |

| | |
|---|---|
| Mobile Cloud Computing(MCC), Offloading in MCC, Load balancing, Crowd sensing, Trust management in MCC; Introduction to Fog, Edge and Dew Computing, Algorithm and applications of Fog, Edge and Dew computing. | |

| **COURSE OUTCOMES** |
|---|
| **After completion of the course, students would be able to:** |
| CO1: Explain the core concepts of the cloud computing paradigm: how and why this paradigm shift came about, the characteristics, advantages and challenges brought about by the various models and services in cloud computing, Cloud Architecture and Services. |
| CO2: Illustrate the fundamental concepts of virtualization, hypervisors and security issues in Cloud Computing. |
| CO3: Develop their understanding about the Advances in Cloud Technologies and emerging areas of research. |

**Textbooks/References:**


1. Anthony T.Velte, Toby J.Velte and Robert E, Cloud Computing – A Practical Approach, TMH 2010
2. Michael Miller, Cloud Computing – Web based Applications, Pearson Publishing, 2011
3. Cloud Computing Explained: Implementation Handbook for Enterprises, John Rhoton, Publication Date: November 2, 2009
4. Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance (Theory in Practice), Tim Mather, ISBN-10: 0596802765,O'Reilly Media, September 2009
5. Cloud Computing Bible, Barrie Sosinsky, Wiley-India, 2010
6. Cloud computing for dummies- Judith Hurwitz , Robin Bloor , Marcia Kaufman ,Fern Halper, Wiley Publishing, Inc, 2010
7. Cloud Computing (Principles and Paradigms), Edited by Rajkumar Buyya, James Broberg, Andrzej Goscinski, John Wiley & Sons, Inc. 2011
8. Cloud Computing: Principles, Systems and Applications, Editors: Nikos Antonopoulos, Lee Gillam, Springer, 2012
9. Cloud Security: A Comprehensive Guide to Secure Cloud Computing, Ronald L. Krutz, Russell Dean Vines, Wiley-India, 2010

**OEC-CS801**

**Cyber law & Ethics**

| | |
|---|---|
| **CourseCode** | OEC-CS801 |
| **CourseName** | Cyber law & Ethics |
| **Credits** | 3 |
| **Pre-Requisites** | Computer Network |

**COURSE OBJECTIVES**

Total Number of Lectures: 36

| |
|---|
| • To introduce the cyber world , ethics in cyber world, and cyber law in general |
| • To explain about the various facets of cyber crimes |
| • To educate about the regulation of cyber space at national and international level |

| **LECTURE WITHBREAKUP** | **NO.OF LECTURES** |
|---|---|
| **Unit1:** Applied Ethics<br>What ethics is and is not, differences between laws and ethics, ethical viewpoints, natural rights, fairness (Justice), Ethical decision making process | 6 |
| **Unit 2**<br>An Overview of Cyber Law, Introduction about the cyber space, Regulation of cyber space – introducing cyber law, Scope of Cyber laws – e-commerce; online contracts; IPRs (copyright, trademarks and software patenting); e-taxation; e-governance and cyber crimes, Cyber law in India with special reference to Information Technology Act, 2000 | 6 |
| **Unit 3:** Domestic Legal Regime – Cyber Law in India<br>Information Technology Act, 2000 – Digital Signature; E-Governance; Regulation of Certifying Authorities; Duties of Subscribers; Penalties and Adjudications; Offences under the Act; Making of Rules and Regulations etc. | 6 |

| | |
|---|---|
| **Unit 4:** Cyber Crimes:<br>Introduction – computer crime and cyber crimes; Classification of cyber crimes.Cyber crime and Related Concepts, Distinction between cyber crime and conventional crimes, Reasons for commission of cyber crime, Cyber forensic, Cyber criminals and their objectives, Kinds of cyber crimes – cyber stalking; cyber pornography; forgery and fraud; crime related to IPRs; Cyber terrorism etc. | 10 |
| **Unit 5:** Regulation of cyber crimes<br>Issues relating to Investigation , Issues relating to Jurisdiction, Issues relating to Evidence, Relevant provisions under Information Technology Act, 2000, Indian Penal Code, Pornography Act and Evidence Act etc. | 5 |
| **Unit 6:**<br>Recent development in cyber crime & amendment in cyber law. | 3 |

**COURSEOUTCOMES**

After completion of course, students would be able to:

| |
|---|
| 1. Identify and analyze statutory, regulatory, constitutional, and organizational laws that affect the information technology professional. |
| 2. Locate and apply case laws and common laws to current legal dilemmas in the technology field. |
| 3. Apply diverse viewpoints to ethical dilemmas in the information technology field and recommend appropriate actions. |

**References:**

1. Cyber security by Nina Gobole & Sunit Belapune; Pub: Wiley India.
2. https://www.meity.gov.in/content/cyber-laws

- **OEC, OPEN ELECTIVE IV, CREDIT 3, Hrs. 3**
- **Project, PROJ-CS891, CREDIT 6, Hr 12**