

FACULTY OF ENGINEERING
Scheme of Instruction & Examination
(AICTE Model Curriculum for the Academic Year 2019-2020)

and

Syllabi

B.E. III and IV Semester

of

Four Year Degree Programme

in

Computer Science and Engineering

(With effect from the academic year 2019–2020)

(As approved in the faculty meeting held on 25-06-2019)



Issued by

Dean, Faculty of Engineering

Osmania University, Hyderabad – 500 007

2019

**SCHEME OF INSTRUCTION & EXAMINATION
B.E. (Computer Science and Engineering) III – SEMESTER**

S. No.	Course Code	Course Title	Scheme of Instruction				Scheme of Examination			Credits
			L	T	P/D	Contact Hrs/Wk	CIE	SEE	Duration in Hrs	
Theory Courses										
1	MC112CE	Environmental Science	2	-	-	2	30	70	3	-
2	MC113PY	Essence of Indian Traditional Knowledge	2	-	-	2	30	70	3	-
3	HS204ME	Operations Research	3	-	-	3	30	70	3	3
4	BS206BZ	Biology for Engineers	3	-	-	3	30	70	3	3
5	ES214EC	Basic Electronics	3	-	-	3	30	70	3	3
6	ES216EC	Digital Electronics	3	-	-	3	30	70	3	3
7	PC221CS	Data Structures and Algorithms	3	-	-	3	30	70	3	3
8	PC222CS	Discrete Mathematics	3	-	-	3	30	70	3	3
9	PC223CS	Programming Languages	3	-	-	3	30	70	3	3
Practical/ Laboratory Courses										
10	ES251EC	Basic Electronics Lab	-	-	2	2	25	50	3	1
11	PC252CS	Data Structures and Algorithms Lab	-	-	2	2	25	50	3	1
12	PC253CS	Advanced Computer Skills Lab	-	-	2	2	25	50	3	1
			25	-	06	31	345	780		24

HS: Humanities and Social Sciences

BS: Basic Science

ES: Engineering Science

MC: Mandatory Course

PC: Professional Core

L: Lecture

T: Tutorial

P: Practical

D: Drawing

CIE: Continuous Internal Evaluation

SEE: Semester End Evaluation (Univ. Exam)

PY: Philosophy

BZ: Biology/ Life Sciences

CE: Civil Engineering

CS: Computer Science and Engineering

EC: Electronics and Communication Engineering

ME: Mechanical Engineering.

Note:

- Each contact hour is a clock hour
- The duration of the practical class is two hours, however it can be extended wherever necessary, to enable the student to complete the experiment.
- All mentioned **Mandatory Courses** for BE (All Branches) should be offered either in I – Semester or II – Semester only **from the academic year 2019-2020**.
- For those of the students admitted in BE (All Branches) during the academic year 2018-2019 the Mandatory Courses were not offered during the I – Semester or II – Semester may be compulsorily offered either in III – Semester or IV – Semester **for the academic year 2019-2020 only**.

Course Code	Course Title				Core/Elective		
MC112CE	Environmental Science				Mandatory		
Prerequisite	Contact Hours per Week				CIE	SEE	Credits
	L	T	D	P			
-	2	-	-	-	30	70	-

Course Objectives

- To create awareness and impart basic knowledge about the environment and its allied problems.
- To know the functions of ecosystems.
- To understand importance of biological diversity.
- To study different pollutions and their impact on environment.
- To know social and environment related issues and their preventive measures.

Course Outcomes

After completing this course, the student will be able to:

1. Adopt environmental ethics to attain sustainable development.
2. Develop an attitude of concern for the environment.
3. Conservation of natural resources and biological diversity.
4. Creating awareness of Green technologies for nation's security.
5. Imparts awareness for environmental laws and regulations.

UNIT-I

The Multidisciplinary Nature of Environmental Studies: Definition, scope and importance, need for public awareness.

Natural Resources: Water Resources – Use and over utilization of surface and ground water, flood, drought, conflicts over water, Dams: Benefits and Problems. Food Resources –World Food Problems, effects of modern agriculture, fertilizer-pesticides problems, water logging, salinity, Forest Resources –Use and over exploitation, deforestation & its effect on tribal people. Land Resources –Land Degradation, environmental effect of mining, man induced landslides, soil erosion and desertification. Energy Resources –Growing energy needs, Renewable and Non-renewable energy resources.

UNIT-II

Ecosystems: Concept of an ecosystem, Structure and function of an ecosystem, Producers, consumers and decomposers, Energy flow in ecosystem, food chains, ecological pyramids, ecological succession, types of ecosystems (marine, pond, river, forest, grassland, desert)

UNIT-III

Biodiversity: Levels of Biodiversity, Bio-geographical classification of India, Value of biodiversity, Threats to biodiversity, endangered and endemic species of India, Conservation of biodiversity, global and national efforts.

UNIT-IV

Environmental Pollution: Definition, Causes, effects and control measures of air pollution, water pollution, soil pollution, noise pollution, thermal pollution, solid waste management.

Environment Protection Act: Air, water, forest and wildlife Acts, issues in the enforcement of environmental legislation.

UNIT-V

Social Issues and the Environment: Watershed management and environmental ethics. Climate change, global warming, acid rain, ozone layer depletion.

Environmental Disaster Management: Types of disasters, impact of disasters on environment, infrastructure, and development. Basic principles of disaster mitigation, disaster management, and methodology. Disaster management cycle and disaster management in India.

Field Work:

- Visit to a local area to document environmental issues- agricultural area/ pond/lake/terrestrial ecosystem
- Visit to a local polluted area- market/slum area/Industrial area/traffic area

Suggested Readings:

1. A.K. De, *Environmental Chemistry*, Wiley Eastern Ltd.
2. E.P. Odum, *Fundamentals of Ecology*, W.B. Saunders Co., USA.
3. M.N. Rao and A.K. Datta, *Waste Water Treatment*, Oxford and IBK Publications.
4. Benny Joseph, *Environmental Studies*, Tata McGraw Hill, 2005.
5. V.K. Sharma, *Disaster Management*, National Centre for Disaster Management, IPE, 1999.

Course Code	Course Title				Core/Elective		
MC113PY	Essence of Indian Traditional Knowledge				Mandatory		
Prerequisite	Contact Hours per Week				CIE	SEE	Credits
	L	T	D	P			
-	2	-	-	-	30	70	-

Course Objectives

The course will introduce the students to

- To get a knowledge in Indian Philosophical Foundations.
- To Know Indian Languages and Literature and the fine arts in India & Their Philosophy.
- To explore the Science and Scientists of Medieval and Modern India

Course Outcomes

After successful completion of the course the students will be able to

1. Understand philosophy of Indian culture.
2. Distinguish the Indian languages and literature among difference traditions.
3. Learn the philosophy of ancient, medieval and modern India.
4. Acquire the information about the fine arts in India.
5. Know the contribution of scientists of different eras.
6. The essence of Yogic Science for Inclusiveness of society.

UNIT – I

Introduction to Indian Philosophy: Basics of Indian Philosophy, culture, civilization, culture and heritage, general characteristics of culture, importance of culture in human literature, Indian culture, Ancient Indian, Medieval India, Modern India.

UNIT – II

Indian Philosophy & Literature: Vedas Upanishads, schools of Vedanta, and other religion Philosophical Literature. Philosophical Ideas the role of Sanskrit, significance of scriptures to current society, Indian Philosophies, literature of south India.

Indian languages and Literature-II: Northern Indian languages & Philosophical & cultural & literature.

UNIT – III

Religion and Philosophy: Religion and Philosophy in ancient India, Religion and Philosophy in Medieval India, Religious Reform Movements in Modern India (selected movements only)

UNIT – IV

Indian Fine Arts & Its Philosophy (Art, Technology & Engineering): Indian Painting, Indian handicrafts, Music, divisions of Indian classic music, modern Indian music, Dance and Drama, Indian Architecture (ancient, medieval and modern), Science and Technology in Indian, development of science in ancient, medieval and modern Indian.

UNIT – V

Education System in India: Education in ancient, medieval and modern India, aims of education, subjects, languages, Science and Scientists of Ancient India, Scientists of Medieval India, Scientists of Modern India. The role Gurukulas in Education System, Value based Education.

Suggested Readings:

1. Kapil Kapoor, “Text and Interpretation: The India Tradition”, ISBN: 81246033375, 2005
2. “Science in Samskrit”, Samskrita Bharti Publisher, ISBN-13:978-8187276333,2007

3. NCERT, "Position paper on Arts, Music, Dance and Theatre", ISBN 81-7450-494-X, 2006
4. S. Narain, "Examination in Ancient India", Arya Book Depot, 1993
5. Satya Prakash, "Founders of Sciences in Ancient India", Vijay Kumar Publisher, 1989
6. M. Hiriyanna, "Essentials of Indian Philosophy", Motilal Banarsidass Publishers, ISBN-13: 978-8120810990,2014
7. Chatterjee. S & Dutta "An Introduction to Indian Philosophy"

Course Code	Course Title				Core/Elective		
HS204ME	Operations Research				Core		
Prerequisite	Contact Hours per Week				CIE	SEE	Credits
	L	T	D	P			
-	3	-	-	-	30	70	3

Course Objectives

- Use variables for formulating complex mathematical models in management science, industrial engineering and transportation models.
- Use the basic methodology for the solution of linear programming problems.
- Understand the mathematical tools that are needed to solve optimization problems like Transportation models and Assignment models.
- Understand the replacement models with change in money value considering with time and without time.
- Model a system as a queuing model and compute important performance measures

Course Outcomes

After completing this course, the student will be able to:

1. Prepare the students to have the knowledge of Linear Programming Problem in Operations
2. Research at the end students would be able to understand the concept and develop the models for different applications.
3. Make students understand the concept Replacement models at the end students would able to explain various features and applications of replacement models in real time scenario.
4. Prepare the students to understand theory of Game in operations research at the end students would able to explain application of Game theory in decision making for a conflict
5. Prepare the students to have the knowledge of Sequencing model at the end student would able to develop optimum model for job scheduling.
6. Prepare students to understand Queuing theory concepts and various optimization techniques at the end students would able to develop models for waiting line cases.

UNIT-I

Introduction: Definition and Scope of Operations Research.

Linear Programming: Introduction, Formulation of linear programming problems, graphical method of solving LP problem, simplex method, maximization and minimization, Degeneracy in LPP, Unbounded and, Infeasible solutions.

UNIT-II

Duality: Definition, Relationship between primal and dual solutions, Economic Interpretation, Post optimal of sensitivity analysis, Dual Simplex Method.

UNIT-III

Transportation Models: Finding an initial feasible solution - North West corner method, least cost method, Vogel's Approximation method, Finding the optimal solution, optimal solution by stepping stone and MODI methods, Special cases in Transportation problems - Unbalanced Transportation problem.

Assignment Problems: Hungarian method of Assignment problem, Maximization in Assignment problem, unbalanced problem, problems with restrictions, travelling salesman problems.

UNIT-IV

Replacement Models: Introduction, replacement of items that deteriorate ignoring change in money value, replacement of items that deteriorate considering change in money value with time, replacement of items that fail suddenly - Individual replacement policy, Group replacement policy.

Game Theory: Introduction, 2 person zero sum games, Maximin - Minimax principle, Principle of Dominance, Solution for mixed strategy problems, Graphical method for 2 x n and m x 2 games.

UNIT-V

Sequencing Models: Introduction, General assumptions, processing n jobs through 2 machines, processing 'n' jobs through m machines, Processing 2 jobs through m machines

Queuing Theory: Introduction, single channel - Poisson arrivals - exponential service times with infinite population & finite population, Multi-channel - Poisson arrivals - Exponential service times with infinite population.

Introduction to Optimization Techniques: Single objective & Multi objective optimization Techniques like G.A, NSGA, P.Q.O & MPSO Techniques.

Suggested Readings:

1. Hamdy, A. Taha, Operations Research-An Introduction, Sixth Edition, Prentice Hall of India Pvt. Ltd., 1997.
2. S.D. Sharma, Operations Research, Kedarnath, Ramnath & Co., Meerut, 2009.
3. Hrvey M. Wagner, Principles of Operations Research, Second Edition, Prentice Hall of India Ltd., 1980.
4. V.K. Kapoor, Operations Research, S. Chand Publishers, New Delhi, 2004.
5. R. Paneer Selvam, Operations Research, Second Edition, PHI Learning Pvt. Ltd., New Delhi, 2008.
6. Data Reconciliation by Prof. Shanker Narasimha

Course Code	Course Title				Core/Elective		
BS206BZ	Biology for Engineers				Core		
Prerequisite	Contact Hours per Week				CIE	SEE	Credits
	L	T	D	P			
-	3	-	-	-	30	70	3

Course Objectives

Gain vivid knowledge in the fundamentals and uses of biology, human system and plant system.

Course Outcomes

After completing this course, the student will be able to:

1. Apply biological engineering principles, procedures needed to solve real-world problems.
2. Understand the fundamentals of living things, their classification, cell structure and biochemical constituents.
3. Apply the concept of plant, animal and microbial systems and growth in real life situations.
4. Comprehend genetics and the immune system.
5. Know the cause, symptoms, diagnosis and treatment of common diseases.
6. Apply basic knowledge of the applications of biological systems in relevant industries.

UNIT-I

Introduction to Life: Characteristics of living organisms, Basic classification, cell theory, structure of prokaryotic and eukaryotic cell, Introduction to Biomolecules: definition, general classification and important functions of carbohydrates, lipids, proteins, vitamins and enzymes.

UNIT-II

Biodiversity: Plant System: basic concepts of plant growth, nutrition, photosynthesis and nitrogen fixation. Animal System: Elementary study of digestive, respiratory, circulatory, excretory systems and their functions. Microbial System: History, types of microbes, economic importance and control of microbes.

UNIT-III

Genetics and Evolution: Theories of evolution and Evidences; cell division—mitosis and meiosis; evidence of laws of inheritance; variation and speciation; nucleic acids as a genetic material; central dogma; Mendel laws, gene and chromosomes.

UNIT-IV

Human Diseases: Definition, causes, symptoms, diagnosis, treatment and prevention of diabetes, cancer, hypertension, influenza, AIDS and Hepatitis. Immunity immunization, antigen – antibody immune response.

UNIT-V

Biology and its Industrial Applications: Transgenic plants and animals, stem cell and tissue engineering, bioreactors, bio pharming, recombinant vaccines, cloning, drug discovery, biological neural networks, bioremediation, biofertilizer, biocontrol, biofilters, biosensors, biopolymers, bioenergy, biomaterials, biochips, basic biomedical instrumentation.

Suggested Readings:

1. A Text book of Biotechnology, R.C. Dubey, S. Chand Higher Academic Publications, 2013
2. Diseases of the Human Body, Carol D. Tamparo and Marcia A. Lewis, F.A. Davis Company, 2011.
3. Biomedical instrumentation, Technology and applications, R. Khandpur, McGraw Hill Professional, 2004

4. Biology for Engineers, Arthur T. Johnson, CRC Press, Taylor and Francis, 2011
5. Cell Biology and Genetics (Biology: The unity and diversity of life Volume I), Cecie Starr, Ralph Taggart, Christine Evers and Lisa Starr, Cengage Learning, 2008
6. Biotechnology Expanding horizon, B.D. Singh, Kalyani Publishers, 2012.

Course Code	Course Title				Core/Elective		
ES214EC	Basic Electronics				Core		
Prerequisite	Contact Hours per Week				CIE	SEE	Credits
	L	T	D	P			
-	3	-	-	-	30	70	3
<p>Course Objectives The objectives of this course is to impart knowledge of</p> <ul style="list-style-type: none"> ➤ To understand the characteristics of diodes and transistor configurations ➤ To understand the design concepts of biasing of BJT and FET ➤ To understand the design concepts of feedback amplifiers and oscillators ➤ To study the design concepts of OP Amp and data converters <p>Course Outcomes After completing this course, the student will be able to:</p> <ol style="list-style-type: none"> 1. Study and analyse the rectifiers and regulator circuits. 2. Study and analyse the performance of BJTs, FETs on the basis of their operation and working. 3. Ability to analyse & design oscillator circuits. 4. Ability to analyse different logic gates & multi-vibrator circuits. 5. Ability to analyse different data acquisition systems 							

UNIT-I

PN Junction Diode: Characteristics, Half wave rectifier, Full wave rectifier, filters, ripple, regulation, TIF and efficiency, Zener diode and Zener diode regulators. CRT construction and CRO applications

UNIT-II

Transistors: BJT construction and working, modes of operation, configurations of BJT (CB, CE, CC), small signal h-parameter model of CE, CE amplifier analysis. Construction and working of JFET, V-I characteristics of JFET.

UNIT-III

Feedback concepts: Types of negative feedback – modification of gain, bandwidth, input and output impedances, applications.

Oscillators: RC Phase shift, Wein bridge, LC and crystal Oscillators (Qualitative treatment only).

UNIT-IV

Operational Amplifier: OP-AMP Block diagram, Ideal OP-AMP, DC and AC Characteristics, Inverting and Non-Inverting Amplifiers, Adder/Subtractor, Integrator, Differentiator.

Logic gate circuits - Introduction to Digital systems- AND, NAND, NOR, XOR gates, Binary half adder, full adder.

UNIT-V

Data Acquisition Systems: Construction and Operation of transducers- Strain gauge LVDT, Thermocouple, Instrumentation systems.

Data Converters: R-2R Ladder DAC, Successive approximation and Flash ADC.

Suggested Readings:

1. Robert Boylestad L. and Louis Nashelsky, *Electronic Devices and Circuit Theory*, PHI, 2007

2. Helfrick D and David Cooper, *Modern Electronic Instrumentation and Measurements Techniques*, 1st edition, Prentice Hall of India, 2006.
3. Salivahanan, Suresh Kumar and Vallavaraj, *Electronic Devices and Circuits*, 2nd edition, Tata McGraw-Hill, 2010.

Course Code	Course Title				Core/Elective		
ES216EC	Digital Electronics				Core		
Prerequisite	Contact Hours per Week				CIE	SEE	Credits
	L	T	D	P			
-	3	-	-	-	30	70	3

Course Objectives

- To learn the principles of digital hardware and support given by it to the software.
- To explain the operation and design of combinational and arithmetic logic circuits.
- To design hardware for real world problems.

Course Outcomes

At the end of this course the students will be able to

1. Understand the design process of digital hardware, use Boolean algebra to minimize the logical expressions and optimize the implementation of logical functions.
2. Understand the number representation and design combinational circuits like adders, MUX etc.
3. Design Combinational circuits using PLDS and write VHDL code for basic gates and combinational circuits.
4. Analyse sequential circuits using flip-flops and design registers, counters.
5. Represent a sequential circuit using Finite State machine and apply state minimization techniques to design a FSM

UNIT – I

Design Concepts: Digital Hardware, Design process, Design of digital hardware. Introduction to logic circuits – Variables and functions, Logic gates and networks. Boolean algebra, Synthesis using gates, Design examples. Optimized implementation of logic functions using K-Map and Quine-McCluskey Tabular method

UNIT – II

Number representation: Addition and Subtraction of signed and unsigned numbers.

Combinational circuit building blocks: Half adder, Full adder, Multiplexers. Decoders. Encoders. Code converters, BCD to 7-segment converter, Arithmetic comparator circuits.

UNIT – III

Design of combinational circuits using Programmable Logic Devices (PLDs): General structure of a Programmable Array Logic (PAL), Programmable Logic Arrays (PLAs), Structure of CPLDs and FPGAs, 2-input and 3-input lookup tables (LUTs)

Introduction to Verilog HDL: Verilog code for basic logic gates, adders, decoders.

UNIT – IV

Sequential Circuits: Basic Latch, Gated SR Latch, gated D Latch, Master-Slave edge triggered flip-flops, T Flip-flop, JK Flip-flop, Excitation tables. Registers, Counters, Verilog code for flip-flops

UNIT – V

Synchronous Sequential Circuits: Basic Design Steps, Finite State machine (FSM) representation using Moore and Mealy state models, State minimization, Design of FSM for Sequence Generation and Detection, Algorithmic State Machine charts.

Suggested Readings:

1. Moris Mano and Michael D Ciletti, Digital Design, Pearson, fourth edition,2008
2. Zvi Kohavi, Switching and Finite Automata Theory, 3rd ed., Cambridge University Press-New Delhi, 2011.
3. R. P Jain, Modern Digital Electronics,4th ed., McGraw Hill Education (India) Private Limited, 2003
4. Ronald J.Tocci, Neal S. Widmer &Gregory L.Moss, “Digital Systems: Principles and Applications,” PHI, 10/e, 2009.
5. Samir Palnitkar, “Verilog HDL A Guide to Digital Design and Synthesis,” 2nd Edition, Pearson Education, 2006.

Course Code	Course Title				Core/Elective		
PC221CS	Data Structures and Algorithm				Core		
Prerequisite	Contact Hours per Week				CIE	SEE	Credits
	L	T	D	P			
-	3	-	-	-	30	70	3

Course Objectives

- To teach the importance of structuring the data for easy access and storage.
- To teach the implementation of various data structures.
- To acquire skills in using generic principles for data representation and manipulation with a view for efficiency, maintainability and code reuse.
- To introduce the basic concepts of advanced data structures.

Course Outcomes

After completing this course, the student will be able to:

1. Understand the importance of abstract data type and implementing the concepts of data structure using abstract data type.
2. Evaluate an algorithm by using algorithmic performance and measures.
3. Distinguish between linear and non-linear data structures and their representations in the memory using array and linked list.
4. Develop applications using Linear and Non-linear data structures.
5. Apply the suitable data structure for a real world problem and think critically for improvement in solutions.
6. Determine the suitability of the standard algorithms: Searching, Sorting and Traversals.

UNIT-I

Introduction to C++ and Algorithms: Object oriented Design, Data Abstraction and Encapsulation, Basics of C++: Program organization in C++, Input/output in C++, Classes and Constructors, Access Modifiers, Dynamic Memory Allocation in C++, Templates in C++, Exception Handling.

Algorithms: Introduction, Algorithm Specifications, Recursive Algorithms, Performance Analysis of an algorithm- Time and Space Complexity, Asymptotic Notations.

UNIT-II

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, Queue ADT and its operations: Linear Queue, Circular Queue, Algorithms and their analysis.

UNIT-III

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.

UNIT-IV

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, Heaps.

UNIT-V

Sorting and Searching: 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, Linear and Binary Search algorithms, and their complexity analysis, Hashing

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

Suggested Readings:

1. “Fundamentals of Data Structures in C++”, Illustrated Edition by Ellis Horowitz, Sartaj Sahni, Dinesh Mehta, 2nd Edition, Universities Press.
2. Algorithms, Data Structures, and Problem Solving with C++”, Illustrated Edition by Mark Allen Weiss, 3rd Edition, Pearson India.
3. “Fundamentals of Data Structures”, Illustrated Edition by Ellis Horowitz, Sartaj Sahni, Computer Science Press.
4. “How to Solve it by Computer”, 2nd Impression by R.G. Dromey, Pearson Education.

Course Code	Course Title				Core/Elective		
PC222CS	Discrete Mathematics				Core		
Prerequisite	Contact Hours per Week				CIE	SEE	Credits
	L	T	D	P			
-	3	-	-	-	30	70	3

Course Objectives

- To Learn mathematical concepts as applied in computer science for solving logical problems.
- To model relationships, analyse data, apply probability concepts and use functions to solve problems.
- To develop the mathematical skills needed for advanced quantitative courses.

Course Outcomes

After completing this course, the student will be able to:

1. Apply Propositional and Predicate logic for a variety of problems in various domains.
2. Understand Set Theory, Venn Diagrams, relations, functions and apply them to Real-world scenarios.
3. Model and solve the real world problems using Generating Functions and Recurrence Relations.
4. To identify the basic properties of graphs and trees and use these concepts to model simple applications.
5. Understand General properties of Algebraic systems and study lattices as partially ordered sets and their applications.
6. Apply the knowledge and skills obtained to investigate and solve a variety of discrete mathematics problems.

UNIT – I

Logic – Sets and Functions – Logic, Propositional equivalences – Predicates and quantifiers – Nested Quantifiers-Sets-Set Operations, Functions.

Algorithms- Integers – Matrices: Algorithms, Complexity of Algorithms. The Integers and Division, Integers and Algorithms, Applications of Number Theory, Matrices.

UNIT – II

Mathematical Reasoning, Induction, and Recursion: Proof Strategy, Sequence and Summation, Mathematical Induction, Recursive Definitions and Structural Induction, Recursive Algorithms.

Counting – Basics, Pigeonhole principle, Permutations and combinations – Binomial Coefficients, Generalized Permutations and combinations, Generating permutations and combinations.

UNIT – III

Discrete Probability: An Introduction to Discrete Probability theory, Expected Value and Variance.

Advanced Counting Techniques: Recurrence relations – Solving Recurrence Relations, - Divide and conquer relations – and Recurrence Relations, Generating function – Inclusion – Exclusion – Applications of Inclusion – Exclusion.

UNIT – IV

Relations: Relations & their Properties, n-ray relations and applications, Representing relations – Closures, equivalence relations, partial orderings.

Graphs: Introduction, Graph terminology, representing Graphs and Graph Isomorphism, Connectivity, Euler and Hamiltonian paths, Shortest path problems, Planar graphs, Graph colouring.

UNIT – V

Trees: Introduction to Trees, Application of Trees, Spanning Trees, Minimum Spanning Trees.

Boolean Algebra: Boolean function, Representing Boolean functions, Logic Gates

Suggested Readings:

1. Kenneth H. Rosen – Discrete Mathematics and its Application – 5th Edition, McGraw Hill, 2003.
2. J. K. Sharma, Discrete Mathematics, Second Edition, Macmillan, 2005.
3. J.P. Tremblay, R. Manohar, Discrete Mathematical Structure with Application to Computer Science, McGraw Hill – 1997.
4. Joel. Mott. Abraham Kandel, T.P. Baker, Discrete Mathematics for Computer Scientist & Mathematicians, Prentice Hall N.J., 2nd Edition, 1986.

Course Code	Course Title				Core/Elective		
PC223CS	Programming Languages				Core		
Prerequisite	Contact Hours per Week				CIE	SEE	Credits
	L	T	D	P			
-	3	-	-	-	30	70	3

Course Objectives

- To briefly describe various programming paradigms.
- To provide conceptual understanding of High level language design and implementation.
- To introduce the power of scripting languages.
- To provide an introduction to formalisms for specifying syntax and semantics of programming languages.
- To provide an exposure to core concepts and principles in contemporary programming languages.
- To analyse and optimize the complexity of the programming languages.

Course Outcomes

After completing this course, the student will be able to:

1. Ability to express syntax and semantics in formal notation.
2. Ability to apply suitable programming paradigm for the application.
3. Gain Knowledge and comparison of the features programming languages
4. program in different language paradigms and evaluate their relative benefits.
5. Identify and describe semantic issues associated with variable binding, scoping rules, parameter passing, and exception handling.
6. Understand the design issues of object-oriented and functional languages.

UNIT- I

Preliminary Concepts: Reasons for studying, concepts of programming languages, Programming domains, Language Evaluation Criteria, influences on Language design, Language categories, Programming Paradigms – Imperative, Object Oriented, functional Programming, Logic Programming. Programming Language Implementation – Compilation and Virtual Machines, programming environments. Syntax and Semantics: general Problem of describing Syntax and Semantics, formal methods of describing syntax - BNF, EBNF for common programming languages features, parse trees, ambiguous grammars, attribute grammars, denotational semantics and axiomatic semantics for common programming language features.

UNIT- II

Data types: Introduction, primitive, character, user defined, array, associative, record, union, pointer and reference types, design and implementation uses related to these types. Names, Variable, concept of binding, type checking, strong typing, type compatibility, named constants, variable initialization. Expressions and Statements: Arithmetic relational and Boolean expressions, Short circuit evaluation mixed mode assignment, Assignment Statements, Control Structures – Statement Level, Compound Statements, Selection, Iteration, Unconditional Statements, guarded commands.

UNIT-III

Subprograms Blocks and Fundamentals of sub-programs: Scope and lifetime of variable, static and dynamic scope, Design issues of subprograms and operations, local referencing environments, parameter passing methods, overloaded sub-programs, generic sub-programs, parameters that are subprogram names, design issues for functions user defined overloaded operators, co routines.

UNIT- IV

Abstract types: Data Abstractions and encapsulation, introductions to data abstraction, design issues, language examples, C++ parameterized ADT, object oriented programming in small talk, C++, Java, C#, Ada 95 Concurrency: Subprogram level concurrency, semaphores, monitors, message passing, Java threads, C# threads. Exception handling: Exceptions, exception Propagation, Exception handler in Ada, C++ and Java. Logic Programming Language: Introduction and overview of logic programming, basic elements of prolog, application of logic programming.

UNIT- V

Functional Programming Languages: Introduction, fundamentals of FPL, LISP, ML, Haskell, application of Functional Programming Languages and comparison of functional and imperative Languages. Scripting Language: Pragmatics, Key Concepts, Case Study: Python – Values and Types, Variables, Storage and Control, Bindings and Scope, Procedural Abstraction, Data Abstraction, Separate Compilation, Module Library.

Suggested Readings:

1. Concepts of Programming Languages Robert W. Sebesta 8/e, Pearson Education, 2008.
2. Programming Language Design Concepts, D. A. Watt, Wiley dreamtech, rp-2007
3. Programming Languages, 2nd Edition, A.B. Tucker, R.E. Noonan, TMH.
4. Programming Languages, K. C. Loudon, 2nd Edition, Thomson, 2003.
5. LISP, Patric Henry Winston and Paul Horn, Pearson Education.
6. Programming in Prolog, W.F. Clocksin, & C.S. Mellish, 5th Edition, Springer.
7. Programming Python, M. Lutz, 3rd Edition, O'reilly, SPD, rp-2007.
8. Core Python Programming, Chun, II Edition, Pearson Education, 2007.
9. Guide to Programming with Python, Michael Dawson, Thomson, 2008

Course Code	Course Title					Core/Elective	
PC251EC	Basic Electronics Lab					Core	
Prerequisite	Contact Hours per Week				CIE	SEE	Credits
	L	T	D	P			
-	-	-	-	2	25	50	1

Course Objectives

- To understand the characteristics of diodes and transistor configurations
- To understand the design concepts of biasing of BJT and FET
- To understand the design concepts of feedback amplifiers and oscillators
- To study the design concepts of OP Amp and data converters

Course Outcomes

After completing this course, the student will be able to:

1. Ability to design diode circuits & understand the application of Zener diode.
2. Ability to analyse characteristics of BJTs & FETs.
3. Ability to understand the different oscillator circuits.
4. Ability to understand operation of HWR & FWR circuits with & without filters.
5. Ability to design Analog-to-Digital converters & Digital-to-Analog converters.

List of Experiments:

1. CRO-Applications, Measurements of R, L and C using LCR meter, Colour code method and soldering practice.
2. Characteristics of Semiconductors diode (Ge, Si and Zener)
3. Static Characteristics of BJT-Common Emitter
4. Static Characteristics of BJT-Common Base
5. Static Characteristics of FET
6. RC-Phase Shift Oscillator
7. Hartley and Colpitts Oscillators
8. Common Emitter Amplifier
9. Astable Multivibrator
10. Full-wave rectifier with and without filters using BJT
11. Operational Amplifier Applications
12. Strain Gauge Measurement
13. Analog-to-Digital and Digital to Analog Converters

Suggested Readings:

1. Maheshwari and Anand, *Laboratory Experiments and PSPICE Simulations in Analog Electronics*, 1st edition, Prentice Hall of India, 2006.
2. David Bell A., *Laboratory Manual for Electronic Devices and Circuits*, Prentice Hall of India, 2001.

Course Code	Course Title				Core/Elective		
PC252CS	Data Structures and Algorithm Lab				Core		
Prerequisite	Contact Hours per Week				CIE	SEE	Credits
	L	T	D	P			
-	-	-	-	2	25	50	1

Course Objectives

- Design and construct simple programs by using the concepts of structures as abstract data type.
- To have a broad idea about how to use pointers in the implement of data structures.
- To enhance programming skills while improving their practical knowledge in data structures.
- To strengthen the practical ability to apply suitable data structure for real time applications.

Course Outcomes

After completing this course, the student will be able to:

1. Implement the abstract data type and reusability of a particular data structure.
2. Implement linear data structures such as stacks, queues using array and linked list.
3. Understand and implements non-linear data structures such as trees, graphs.
4. Implement various kinds of searching, sorting and traversal techniques and know when to choose which technique.
5. Understanding and implementing hashing techniques.
6. Decide a suitable data structure and algorithm to solve a real world problem.

Programming Exercise using C++:

1. C++ Programs to implement: Classes, Constructors, Inheritance, Polymorphism, Dynamic Memory Allocation, Class Templates, Exception Handling.
2. Implementation of Stacks, Queues (using both arrays and linked lists).
3. Implementation of Singly Linked List, Doubly Linked List and Circular List.
4. Implementation of Infix to Postfix conversion and evaluation of postfix expression.
5. Implementation of Polynomial arithmetic using linked list.
6. Implementation of Linear search and Binary Search
7. Implementation of Hashing Technique
8. Implementation of Binary Tree and Binary tree traversal techniques (inorder, preorder, postorder, level-order)
9. Implementation of Binary search tree and its operations
10. Implementation of Insertion Sort, Selection Sort, Bubble Sort, Merge Sort, Quick Sort, Heap Sort
11. Implementation of operations on AVL trees.
12. Implementation of Graph Search Methods.

Note: It is recommended to use a debugging tool to debug the programs.

Course Code	Course Title				Core/Elective		
PC253CS	Advanced Computer Skills Lab				Core		
Prerequisite	Contact Hours per Week				CIE	SEE	Credits
	L	T	D	P			
-	-	-	-	2	25	50	1
Course Objectives <ul style="list-style-type: none"> ➤ Introducing a new object oriented programming ➤ Enabling students to learn Big Data, Machine Learning etc. ➤ Preparing students to cope up with new Market tendencies ➤ To learn programs in MATLAB environment ➤ To handle Functions, Polynomials by using MATLAB commands ➤ Ability to solve any Mathematical functions ➤ To learn Mathematical Modelling in a new approach ➤ To plot Graphics (2-D) easily and effectively Course Outcomes After completing this course, the student will be able to: <ol style="list-style-type: none"> 1. Implement basic syntax in python. 2. Analyse and implement different kinds of OOP concept in real world problems. 3. Implement MATLAB operations and graphic functions. 							

List of Programming Exercises:

1. Python Variables, Executing Python from the Command Line, Editing Python Files, Python Reserved Words.
2. Comments, Strings and Numeric Data Types, Simple Input and Output.
3. Control Flow and Syntax, Indenting, if Statement, Relational Operators, Logical Operators, Bit Wise Operators, while Loop, break and continue, for Loop, Lists, Tuples, Sets, Dictionaries.
4. Functions: Passing parameters to a Function, Variable Number of Arguments, Scope, Passing Functions to a Function, Mapping Functions in a Dictionary, Lambda, Modules, Standard Modules.
5. OOP concepts: Classes, File Organization, Special Methods, Inheritance, Polymorphism, Special Characters, Character Classes, Quantifiers, Dot Character, Greedy Matches, Matching at Beginning or End, Match Objects, Compiling Regular Expressions.
6. MATLAB Menus, Toolbars, Computing with MATLAB, Script Files and the Editor/Debugger, MATLAB help System.
7. MATLAB controls: Relational Logical Variables. Conditional Statements: if – else – elseif, switch 2 10. Loops: for – while – break, continue. User-Defined Functions.
8. Arrays, Matrices and Matrix Operations Debugging MATLAB Programs. Working with Data Files, and Graphing Functions: XY Plots – Sub-plots.

Suggested Readings:

1. Mark Summerfield, "Programming in Python
2. A Complete introduction to the Python Language", Addison-Wesley Professional, 2009.
3. Martin C. Brown, "PYTHON: The Complete Reference", McGraw-Hill, 2001.
4. W.J. Palm III, Introduction to MATLAB 7 for Engineers, McGraw-Hill International Edition, 2005.
5. Wesley J Chun, "Core Python Applications Programming", Prentice Hall, 2012.
6. Allen B Downey, "Think Python", O'Reilly, 2012.
7. Stormy Attaway, "MATLAB: A Practical Introduction to Programming and Problem Solving".3rd Edition.