Choice Based Credit System (CBCS)

Name of the Programme (UG): B.E

Syllabus for III - Semester and IV - Semester

With effect from 2017 - 2018

Specialization /Branch: Information Technology

Chaitanya Bharathi Institute of Technology (A)
Chaitanya Bharathi (P.O), Gandipet
Hyderabad-500075, Telangana State.

CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY (A)
Choice Based Credit System
B.E (Information Technology)
SEMESTER – III

S.No.	Course Code	Title of the Course	Scheme of Instruction		Scheme of Examination			Credits
			Hours per week		Duration of SEE in Hours	Maximum Marks		
			L/T	P/D		CIE	SEE	
THEORY								
1	16MT C05	Engineering Mathematics-III	3	-	3	30	70	3
2	16IT C01	Discrete Structures and Applications	3	-	3	30	70	3
3	16IT C02	Data Structures and Algorithms	3/1	-	3	30	70	4
4	16IT C03	Object Oriented Programming	3/1	-	3	30	70	4
5	16IT C04	Digital Electronics and Logic Design	3	-	3	30	70	3
PRACTICALS								
6	16IT C05	Data Structures and Algorithms Lab	-	3	3	25	50	2
7	16IT CO6	Object Oriented Programming Lab	-	3	3	25	50	2
8	16IT C07	Mini Project - I	-	2	-	50	-	1
9	16EG C03	Soft Skills and Employability Enhancement Lab	-	2	3	15	35	1
Total			**17**	**10**	**-**	**265**	**485**	**23**

L: Lecture T: Tutorial D: Drawing P: Practical

CIE-Continuous Internal Evaluation SEE-Semester End Examination

Assessment Procedures for Awarding Marks

The distribution of marks is based on CIE by concerned teacher and the Semester end examination shall be as follows:

Course (in terms of credits)	CIE	Semester end Examination(Marks)	Remarks	Duration of SemesterEnd Examination
Three(3) Credits/ Four(4) Credits	30*	70**	Theory Course/ Engg.Graphics	3 Hours
Two(2) Credits	20*	50***	Theory	2 Hours
Two(2) Credits	25	50	Lab Course/ Workshop	3 Hours
One(1) Credit	15	35	Lab Course	2 Hours
Two(2) Credits	50	—-	Project Seminar/ Seminar	——
Six(6) Credits	50	100	Project	Viva
One(1) Credit	—-	50***	Environmental Studies,Professional Ethics and Human values	2 Hours
One(1) Credit	50		Mini Project	——-

CIE: Continuous Internal Evaluation

*Out of 30/20 sessional marks(CIE), 10/5 marks are allotted for slip-tests (Three slips test will be conducted, each of 10/5 marks, best two average is considered) and the remaining 20/15 marks are based on the average of two tests, weightage for each test is 20/15 marks.

**The question paper will be in two parts, Part-A and Part-B. Part A is for Ten(10) questions and is compulsory, covers the entire syllabus, and carries 20 marks. Part-B carries 50 marks and covers all the units of the syllabus (student has to answer five out of seven questions).

***The question paper will be in two parts, Part-A and Part-B. Part A is for Ten(10) questions and is compulsory, covers the entire syllabus, and carries 15 marks. Part-B carries 35 marks and covers all the units of the syllabus (student has to answer five out of seven questions).

Note: A course that has CIE (sessional marks) but no semester end examination as per scheme, is treated as Pass/Fail for which pass marks are 50% of CIE.

A candidate has earned the credits of a particular course, if he/she secures not less than the minimum marks/grade as prescribed. Minimum pass marks for theory course is 40% of total marks i.e., CIE plus semester end examinations where as for the lab course/project is 50%.

16MT C05

ENGINEERING MATHEMATICS-III

Instruction	3 Hours per week
Duration of End Examination	3 Hours
Semester End Examination	70 Marks
CIE	30 Marks
Credits	3

Course objectives:

1. To study the expansion of functions in various intervals.
2. To form P.D.E and to find its solution.
3. To solve Wave, Heat & Laplace equations.
4. To learn Differentiation and Integration of complex valued functions.
5. To evaluate Complex Integration.
6. To evaluate Real definite integrals.

Course outcomes:

After successful completion of this course, student will be able to

1. Expand functions in the given intervals.
2. Solve linear and non linear PDEs.
3. Solve one-dimension, two-dimension, Heat steady state equations and also one-dimension wave equation.
4. Solve problems on Analytic functions, Cauchy's theorem and Cauchy's integral formula.
5. Expand functions by using Taylor's and Laurent's series.
6. Solve Real and Complex integrals by using Cauchy Theorems.

UNIT – I

Fourier series: Definition of Periodic, Single valued, finite maxima and minima of functions. Euler's Formulae, Dirichlets Conditions for Fourier expansion, Functions having points of discontinuity, Change of interval, Expansion of odd and even functions, Half-range sine series and cosine series.

UNIT-II:

Partial differential equations: Formation of partial differential equations by eliminating the arbitrary constants or arbitrary functions, solutions of linear partial differential equation of first order by using Lagrange's Method, solution of Non-linear partial differential equations of first order by using standard types, Charpit's Method.

UNIT - III

Applications of Partial differential equations: Solution of partial differential equations by using method of separation of variables, solution of vibration of a stretched string (1D-Wave equation), one dimensional heat equation, Two dimensional heat equation under steady state conditions.

UNIT - IV

Theory of Complex variables: Analytic functions, Cauchy Riemann equations (Cartesian and polar forms), construction of Analytic functions by using Milne-Thomson's method. Harmonic function. Complex line integrals, Cauchy's theorem, Cauchy's Integral formula and its derivatives and problems related to the above theorems.

UNIT - V

Expansion of functions, Singularities & Residues: Taylor's and Laurent's series Expansions (Only statements). Zeros, types of singularities, Residues and Cauchy's Residue theorem, Evaluation of real integrals by Cauchy's residue theorem. Evaluation of improper real integrals of the type: $\int_{-\infty}^{\infty} f(x)dx$ Where f(x) has no poles on real axis and $\int_0^{2\pi} f(\sin\theta, \cos\theta)d\theta$.

Text Books:

1. B.S. Grewal, "Higher Engineering Mathematics", 43rd Edition, Khanna Publishers, 2015.
2. M.D. Raisinghania, "Advanced Differential equations", 7th edition, S Chand publishers, 2013.
3. J. W. Brown and R. V. Churchill, "Complex Variables and Applications", 7th edition, McGraw Hill publishers, 2003.

Suggested Reading:

1. N P Bali and Manish Goyal, "A Text Book of Engineering Mathematics", 9th Edition, Laxmi publishers, 2016.
2. Alan Jeffrey, "Mathematics for Engineers and Scientists", 6th Edition, Chapman & Hall/CRC publishers, 2013.
3. A R Vasistha and R K Gupta, "Integral transforms", Krishna prakashan publishers, 2004.
4. R.K.Jain & S.R.K.Iyenger, "Advanced Engineering Mathematics", 3rd edition, Narosa Publications, 2007.

16IT C01

DISCRETE STRUCTURES AND APPLICATIONS

Instruction	3 Hours per week
Duration of Semester End Examination	3 Hours
Semester End Examination	70 Marks
CIE	30 Marks
Credits	3

Course Objectives:

1. To introduce mathematical concepts like sets, functions, logic and to apply them in solving logic oriented problems.
2. To solve problems using graphs to model relationships, analyse data, apply probability concepts and recursive functions.
3. Develop mathematical concepts and techniques that serve as a preparation for more advanced quantitative courses.

Course Outcomes:

After successful completion of this course, student will be able to

1. Symbolize the given sentence using predicate logic and verify the given predicate formula and validity of the argument using universal specification and generalization and equivalence rules.
2. Understand basics of counting, apply permutations and combinations to handle different types of objects.
3. Describe and use recursively-defined relationships to solve problems using generating functions.
4. Analyze semi group, monoidgroup and abelian group with suitable examples and appreciate group theory applications in computer arithmetic.
5. Model problems in Computer Science using graphs and trees.
6. Demonstrate different traversal methods for trees and graphs.

Prerequisites:

1. Elementary Algebra.
2. Introductory computer science course with C and C++.

UNIT – I

Logic – Sets and Functions: Logic, Propositional equivalences – Predicates and Quantifiers – Nested Quantifiers-Rules of Inference-Sets-Set Operations, Functions.

Integers: The Integers and Division, Integers and Algorithms, Applications of Number Theory.

UNIT – II

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

Counting: Basics of Counting, Pigeonhole Principle, Permutations and Combinations– Binomial Coefficients, Generalized Permutations and Combinations, Generating Permutations and Combinations.

UNIT – III

Advanced Counting Techniques: Recurrence Relations, Solving Linear Recurrence Relations, Divide and Conquer Algorithms and Recurrence Relations, Generating Functions, Inclusion–Exclusion, Application of Inclusion – Exclusion.

Relations: Relations & their Properties, N-ary Relations and Applications, Representing Relations, Closures of Relations, Equivalence Relations, Partial Orderings.

UNIT – IV

Algebraic Structures: Algebraic System - General Properties, semi groups, Monoids, Homomorphism, Groups, Residue arithmetic, group codes and their applications.

UNIT –V

Graphs: Graphs and Graph Models, Graph Terminology, Representing Graphs and Graph Isomorphism, Connectivity, Euler and Hamilton Paths, Shortest Path Problems, Planar Graphs, Graph Coloring.

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

Text Books:

1. Kenneth H Rosen, “Discrete Mathematics and its applications”, Sixth Edition, McGraw Hill, 2006.
2. R.K. Bishit, H.S. Dhami, “Discrete Mathematics” Oxford University Press 2015.

Suggested Reading:

1. J.P.Trembly, R.Manohar, “Discrete Mathematical Structure with Application to Computer Science”, McGraw- Hill – 1997.
2. J. K. Sharma, “Discrete Mathematics”, Second edition, Macmillan, 2005.
3. Joel. Mott.AbrahamKandel, T.P.Baker, “Discrete Mathematics for Computer Scientist &Mathematicans”, Prentice Hail N.J.,
4. C.L. Liu, “Elements of Discrete mathematics”, McGraw-Hill, Third Edition.
5. U.S. Gupta, “Discrete Mathematical Structures”, Pearson, 2014.

16IT C02

DATA STRUCTURES AND ALGORITHMS

Instruction	3L+1T Hours per week
Duration of Semester End Examination	3 Hours
Semester End Examination	70 Marks
CIE	30 Marks
Credits	4

Course Objectives:

1. To develop proficiency in the specification, representation of various linear and nonlinear data structures.
2. To discuss applications of data structures.
3. To familiarize with various pattern matching algorithms and hashing.
4. To develop a base for advanced computer science study.

Course Outcomes:

After successful completion of this course, student will be able to

1. Understand the basic data structures arrays and linked lists.
2. Analyse time complexity of both iterative and recursive functions.
3. Define ADT necessary for solving problems based on Stacks and Queues.
4. Develop solutions using binary trees, advanced search trees, tries and graphs.
5. Use hash functions and handle collisions.
6. Understand various kinds of sorting techniques and apply appropriate techniques for solving a given problem.

Prerequisites:

Programming and Problem Solving (16CSC01), Programming Laboratory (16CSC02).

UNIT- I

Arrays, Linked Lists, and Recursion: Using Arrays, Storing Game Entries in an Array, Sorting an Array, Two-Dimensional Arrays, Singly Linked Lists, Implementing a Singly Linked List, Insertion to the Front of a Singly Linked List, Removal from the Front of a Singly Linked List, Implementing a Generic Singly Linked List, Doubly Linked Lists, Insertion into a Doubly Linked List, Removal from a Doubly Linked List, Circularly Linked Lists, Reversing a Linked List, Recursion, Linear Recursion, Binary Recursion, Multiple Recursion, Analysis of Algorithms.

UNIT- II

Stacks, Queues, and Deques: Stacks, The Stack Abstract Data Type, The STL Stack, A C++ Stack Interface, A Simple Array-Based Stack

Implementation, Implementing a Stack with a Generic Linked List, Reversing a Vector Using a Stack, Matching Parentheses and HTML Tags, Queues, The Queue Abstract Data Type, The STL Queue, A C++ Queue Interface, A Simple Array-Based Implementation, Implementing a Queue with a Circularly Linked List, Double-Ended Queues, The Deque Abstract Data Type, The STL Deque, Implementing a Deque with a Doubly Linked List.

List and Iterator ADTs: Lists, Node-Based Operations and Iterators, The List Abstract Data Type, STL Lists, STL Containers and Iterators.

UNIT- III

Trees: General Trees, Tree Definitions and Properties, Tree Functions, A C++ Tree Interface, A Linked Structure for General Trees, Tree Traversal Algorithms, Depth and Height, Preorder Traversal, Postorder Traversal, Binary Trees, The Binary Tree ADT,A C++ Binary Tree Interface, Properties of Binary Trees, A Linked Structure for Binary Trees, A Vector-Based Structure for Binary Trees, Traversals of a Binary Tree, Representing General Trees with Binary Trees.

Strings: Pattern Matching Algorithms, Brute Force, The Boyer-Moore Algorithm, The Knuth-Morris-Pratt Algorithm, Tries, Standard Tries, Compressed Tries, Suffix Tries.

Sorting: Merge-Sort , Divide-and-Conquer, Merging Arrays and Lists, The Running Time of Merge-Sort , Merge-Sort and Recurrence Equations, Quick-Sort, Randomized Quick-Sort, Studying Sorting through an Algorithmic Lens, A Lower Bound for Sorting, Linear-Time Sorting: Bucket-Sort and Radix-Sort, Comparing Sorting Algorithms.

UNIT- IV

Search Trees: Binary Search Trees, Searching, Update Operations, C++ Implementation of a Binary Search Tree, AVL Trees, Update Operations, Splay Trees, Splaying, When to Splay, Amortized Analysis of Splaying, Tree, Multi-Way Search Trees, Update Operations for (2,4) Tree, Red-Black Trees, Update Operations.

Heaps and Priority Queues: The Priority Queue Abstract Data Type, Keys, Priorities, and Total Order Relations, Comparators, The Priority Queue ADT, A C++ Priority Queue Interface, Sorting with a Priority Queue, The STL priority queue Class, Implementing a Priority Queue with a List, Selection-Sort and Insertion-Sort, Heaps, The Heap Data Structure, Complete Binary Trees and Their Representation, Implementing a Priority Queue with a Heap, Bottom-Up Heap Construction.

UNIT- V

Hash Tables: Hash Tables, Bucket Arrays, Hash Functions, Hash Codes, Compression functions, Collision-Handling Schemes, Load Factors and Rehashing.

Graph Algorithms: Graphs, The Graph ADT, Data Structures for Graphs, The Edge List Structure, The Adjacency List Structure, The Adjacency Matrix Structure, Graph Traversals, Depth-First Search, Implementing Depth-First Search, Breadth-First Search, Directed Graphs, Traversing a Digraph, Transitive Closure, Directed Acyclic Graphs, Shortest Paths, Weighted Graphs, Dijkstra's Algorithm, Minimum Spanning Trees, Kruskal's Algorithm, The Prim-Jarn´ýk Algorithm.

Text Books:

1. Michael T. Goodrich, Roberto Tamassia, David M. Mount, "Data Structure and Algorithms in C++", 2nd Edition, John Wiley, 2011.
2. Narasimha Karumanchi, "Data Structures and Algorithms Made Easy: Data Structures and Algorithmic Puzzles", CareerMonk Publications, 2016.

Suggested Reading:

1. Mark Allen Weiss, "Data Structures and Algorithm Analysis in C++", 3rd Edition Addison-Wesley, 2007.
2. Narasimha Karumanchi, "Data Structures and Algorithms for GATE", CareerMonk Publications, 2011.
3. D. Samantha, "Classic Data Structures", Prentice Hall India, 2nd Edition, 2013.

Web Resources:

1. NPTEL Videos: Introduction to data structures and algorithms - http://nptel.ac.in/courses/106102064/1.
2. https://www.cs.usfca.edu/~galles/visualization/Algorithms.html.
3. https://visualgo.net/en.

16IT C03

OBJECT ORIENTED PROGRAMMING

Instruction	3L+1T Hours per week
Duration of Semester End Examination	3 Hours
Semester End Examination	70 Marks
CIE	30 Marks
Credits	4

Course Objectives:

1. To introduce object-oriented concepts and how they are supported by C++.
2. To facilitate students with the skills required to solve problems using object oriented concepts.
3. To impart the knowledge required to develop user interfaces and application environments.

Course Outcomes:

After successful completion of this course, student will be able to

1. Understand the difference between object oriented programming and procedural oriented language in C++.
2. Understand and analyse the basic concepts of Object Oriented Programming.
3. Apply more advanced C++ features such as composition of objects, operator overloads, dynamic memory allocation, inheritance and polymorphism, file I/O, exception handling, etc.
4. Design, write and test programs that make appropriate use of object-oriented facilities, common to many object-oriented languages such as classes, overloading and inheritance.
5. Implement, document, test and debug solutions in C++.
6. Analyze and implement features of object oriented programming to solve real world problems.

Prerequisites:

Programming and Problem Solving(16CSC01).

UNIT I:

Object-Oriented Paradigm- OOPS- A new Paradigm, Evolution of Programming Paradigms, Moving from C to C++ Data Types, Operators and Expressions, Control Flow, Strings Modular Programming with Functions- Function Components, Passing Data to Functions, Function Return Data Type, Parameter Passing, Return by Reference, Default

Arguments, Inline Functions, Function Overloading, Function Templates, Functions with Variable Number of Arguments, Recursive Functions.

UNIT II:

Classes and Objects: Class Specification, Class Objects, Accessing Class Members, Member Functions, Outside Member Functions as Inline, Accessing Member Functions within the Class, Data Hiding, Access Boundary of Objects Revisited, Empty Class, Pointers within a Class, Passing Objects as Arguments, Returning Objects from Functions, Friend Function and Friend Classes, Constant Parameters and Member Functions, Structures and Classes, Static Data Members and Member Functions. **Object Initialization and Clean-up: Constructors—**Parameterised Constructors, Destructor, Order of Construction and Destruction, Constructors with Default Arguments, Nameless Objects, Dynamic Initialization Through Constructors, Constructors with Dynamic Operations, Copy Constructor, Constructor with Two Dimensional Arrays, Constant Objects and Constructor, Static Data Members with Constructors and Destructors.

Dynamic Objects: Pointers to Objects, Array of Objects, Array of Pointers to Objects, Pointers to Objects Members, this Pointer, Self – Referential Classes, Passing Objects Parameters.

UNIT III:

Operator Overloading: Overloadable Operators, Unary Operator Overloading, Operator keyword, Operator Return Values, Nameless Temporary Objects, Limitations of Increment/ Decrement Operators, Binary Operator Overloading, Arithmetic Operators, Concatenation of Strings, Comparison Operators, Assignment Operators, New and Delete Operators, Conversion Between Objects and Basic types and Objects of different classes, Subscript and Assignment Operator overloading, Overloading with Friend Functions.

Inheritance: Derived Class Declaration, Forms of Inheritance, Constructors and Destructors in derived classes, Constructor invocation and data member initialization, Overloaded Member Functions, Types of Inheritances, Abstract classes and virtual base classes.

Exception Handling: Error Handling, Exception handling model, Exception handling constructs, Lists of exceptions, catch all exceptions, exceptions in: Constructors, Destructors, Operator overloaded functions, Inheritance Tree, Class Templates.

UNIT IV:

Virtual Functions: Need for virtual functions, Pointer to derived class objects, definition of virtual functions, Array of pointers to base class objects, Pure virtual functions, Abstract classes, Virtual destructors, Dynamic Binding.

Streams Computation with Console: Introduction, Predefined console streams, Hierarchy of console streams, unformatted and formatted i/o operations, manipulators, stream operators with user defined classes.

UNIT V:

Streams Computation with Files : Introduction, Hierarchy of File stream classes, opening and closing of files, file modes, file pointers and their manipulators, Sequential and Random access to a file, ASCII and Binary files, saving and retrieving of objects, fstream class, Random Access to a File, Error handling during file manipulation, Command line arguments.

Generic Programming with Templates: Function template, Overloaded function templates, Nesting of function calls, Multiple arguments function template, user defined template arguments, Class templates, Inheritance of class templates, class template with overloaded operators.

Text Books:

1. K.R.Venugopal, RajkumarBuyya, "Mastering C++", 2/e, TMH, 2016.
2. Paul Deitel, Harvey Deitel, "How to Program C++", 9th Edition, Pearson, 2013.

Suggested Reading:

1. Bjarne stronusstrup, "The C++ Programming Language", 4/e, Pearson, 2013.
2. SouravSahay, "Object Oriented Programming with C++", 2/e, Oxford University Press.

Web Resources:

1. https://www.tutorialspoint.com/cplusplus/.
2. https://www.programiz.com/cpp-programming.
3. https://www.class-central.com/tag/c++.

16IT C04

DIGITAL ELECTRONICS AND LOGIC DESIGN

Instruction	3 Hours per week
Duration of Semester End Examination	3 Hours
Semester End Examination	70 Marks
CIE	30 Marks
Credits	3

Course Objectives:

1. To familiarize students with the principles of digital Hardware.
2. To explain the operation and design of combinational and arithmetic logic circuits.
3. To facilitate with the skills required to use HDL tools.

Course Outcomes:

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

1. Design complex logic circuits, do simplification, analysis and synthesis.
2. Simulate digital circuits/systems design using VHDL.
3. Understand the principles of different combinational and arithmetic logic designs and VHDL code.
4. Acquire knowledge about the design of Latches and Flip-flops and their Applications.
5. Understand the basic steps of Synchronous Sequential Circuits.
6. Gain knowledge about the behaviour, analysis and synthesis of Asynchronous Sequential Circuits.

Prerequisites:

Elements of Electronics and Communications Engineering (16ECC01),Applied Physics (16PYC02).

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 AND, OR, and NOT Gates, Design examples. Optimized implementation of logic functions – Karnaugh Map, Strategies for minimization, minimizing Product-of-Sum Forms, Incompletely Specified functions. NAND and NOR logic networks, Very High Speed Integrated Circuit Hardware Description Language (VHDL).

UNIT – II

Programmable logic devices: general structure of a Programmable Logic Array (PLA), gate level diagram, schematic diagram, Programmable Array

Logic (PAL) Structure of CPLDs and FPGAs, 2-input and 3-input lookup tables (LUT). Design of Arithmetic-circuits, VHDL for Arithmetic-circuits, Combinational circuit building blocks – Multiplexers, Decoders, Encoders, Code converters, Arithmetic comparison circuits. VHDL for Combinational circuits.

UNIT – III

Basic Latch, Gated SR Latch, Gated D Latch, Master-Slave and Edge-Triggered D Flip-Flops, D Flip-Flops with Clear and Preset. T Flip-flop, JK Flip-flop, Excitation tables and timing diagrams Registers: Shift Register, Parallel-Access Shift Register, Counters: Asynchronous and Synchronous counters, BCD counter, Ring counter, Johnson counter.

UNIT – IV

Synchronous Sequential Circuits – Basic design steps.State-Assignment problem Moore and Mealy state model.State minimization, Finite State Machine (FSM) as an Arbiter Circuit. Algorithmic State Machine (ASM) charts, formal model.

UNIT – V

Asynchronous Sequential Circuits – Behaviour, Analysis, Synthesis, State reduction, State Assignment, examples. Hazards: static and dynamic hazards, Significance of Hazards.

Text Books:

1. Stephen Brown, ZvonkoVranesic, “Fundamentals of Digital Logic with VHDL design”, 2nd Edition, McGraw Hill, 2009.
2. ZVI Kohavi, Switching and Finite Automata Theory, 2ndedition, Tata McGraw Hill, 1995.

Suggested Reading:

1. Jain R.P., “Modern Digital Electronics,” 3rd edition, TMH, 2003.
2. John F. Wakerly, “Digital design Principles & Practices”, 3rdedition, Prentice Hall, 2001.
3. M. Morris Mano, Charles R. Kime, “Logic and Computer Design Fundamentals”, 2nd edition, Pearson Education Asia, 2001.
4. William I Fletcher, “An Engineering Approach to Digital Design”, Eastern Economy Edition, PHI.
5. H.T. Nagle, “Introduction to Computer Logic”, Prentice Hall, 1975.

16IT C05

DATA STRUCTURES AND ALGORITHMS LAB

Instruction	3 Hours per week
Duration of Semester End Examination	3 Hours
Semester End Examination	50 Marks
CIE	25 Marks
Credits	2

Course Objectives:

1. To introduce basic data structures and algorithms.
2. To introduce Non-linear data structures.
3. To familiarise students with graph operations and algorithms.
4. To familiarise students with advanced tree structures like AVL and Tries.

Course Outcomes:

After successful completion of this course, student will be able to

1. Implement various data structures using arrays, linked lists.
2. Develop ADT necessary for solving problems based on Stacks and Queues.
3. Implement binary trees, general tree structures, advanced search trees, heaps, graphs.
4. Implement tries.
5. Implement hash functions and handle collisions.
6. Implement various kinds of sorting techniques and apply appropriate techniques for solving a given problem.

Prerequisites:

Programming and Problem Solving (16CSC01), Programming Laboratory (16CSC02).

List of Programs

1. Define List ADT and implement its operations.
2. Implement Stack ADT and perform arithmetic expression evaluation.
3. Implement Queues, Circular Queues and Deques.
4. Define String ADT and implement Boyer Moore pattern matching algorithm.
5. Implement Tries.
6. Implement the following Sorting Techniques: Insertion Sort, Bubble Sort, Selection Sort, Shell Sort, Merge Sort, Quick Sort and Heap Sort.
7. Construct a Binary Search Tree and implement Tree Traversals.
8. Implement AVL Tree.
9. Implement Hashing with chaining.
10. Implement Graph Traversals.

Text Books:

1. Narasimha Karumanchi, "Data Structures and Algorithms Made Easy: Data Structures and Algorithmic Puzzles", CareerMonk Publications, 2016.
2. Michael T. Goodrich, Roberto Tamassia, David M. Mount, "Data Structure and Algorithms in C++", 2nd Edition, John Wiley, 2011.

Suggested Reading:

1. Narasimha Karumanchi, "Coding Interview Questions", 3rd Edition, CareerMonk Publications, 2016.
2. Mark Allen Weiss, "Data Structures and Algorithm Analysis in C++", 3rd Edition Addison-Wesley, 2007.

Web Resources:

1. NPTEL Videos Introduction to data structures and algorithms - http://nptel.ac.in/courses/106102064/1.
2. https://www.cs.usfca.edu/~galles/visualization/Algorithms.html.
3. https://visualgo.net/en.

16IT C06

OBJECT ORIENTED PROGRAMMING LAB

Instruction	3 Hours per week
Duration of Semester End Examination	3 Hours
Semester End Examination	50 Marks
CIE	25 Marks
Credits	2

Course Objectives:

1. To familiarize students with object-oriented concepts and their implementation in C++.
2. To facilitate students with the skills required to solve problems using object oriented concepts.
3. To impart the knowledge required to write code with good coding practices.

Course Outcomes:

After successful completion of this course, student will be able to

1. Understand the process of writing, compiling and executing programs in C++ using appropriate predefined functions in C++.
2. Implement the object oriented concepts in developing application using C++.
3. Developing applications in C++ using the understanding of Inheritance andpolymorphism.
4. Understand and use exception handling while developing a C++ application.
5. Understand stream I/O, Files and usage of theavailable classes to handle stream objects in C++ language.
6. Develop complex applications by identifying the appropriate features of objectoriented programming to solve real world problems using C++.

List of Programs

1. To implement parameter passing techniques in functions.
2. To create a Class, Objects and illustrate Static members in a class.
3. To illustrate function overloading and inline function, Friend Functions.
4. To implement types of Constructor, Destructor and Array pointers.
5. To implement Method Overloading and Manipulation of strings.
6. To overload Unary Operator and Binary Operator.

7. To Illustrate types of inheritance and Exception handling.
8. Illustrate virtual functions, Pointer to derived class objects, pure virtual functions, Abstract classes and virtual destructors.
9. To implement streams and perform operations on sequential access file and random access file.
10. Illustrate Function Template and Class Template.

Text Books:

1. K.R.Venugopal, Rajkumar Buyya, "Mastering C++", 2/e, TMH, 2016.
2. Paul Deitel , Harvey Deitel, "How to Program C++", 9/e, Pearson, 2013.

Suggested Reading:

1. Bjarne Strousstrup, "The C++ Programming Language", 4/e, Pearson, 2013.
2. Sourav Sahay, "Object Oriented Programming with C++", 2/e, Oxford University Press.

Web Resources:

1. https://www.tutorialspoint.com/cplusplus/.
2. https://www.programiz.com/cpp-programming.
3. https://www.class-central.com/tag/c++.

16IT C07

MINI PROJECT – I

Instruction	2 Hours per week
Duration of Semester End Examination	-
Semester End Examination	-
CIE	50 Marks
Credits	1

Course Objectives:

1. To enable students to learn by doing, to take responsibility of the end product.
2. To develop capability to analyse and solve real world problems with an emphasis on applying/integrating knowledge acquired.

Course Outcomes:

After successful completion of this course, student will be able to

1. Construct innovative solutions.
2. To work in team as well as individuals.
3. To manage time and resources.

The Students are required to implement one of the projects from project exercise given in the suggested readings of the theory subjects. During the implementation of the project, Personnel Software Process (PSP) has to be followed.

Report of the project work has to be submitted for evaluation.

16EG CO3

SOFT SKILLS AND EMPLOYABILITY ENHANCEMENT LAB

Instruction	2 hours per week
Duration of Semester End Examination	2 Hours
Semester End Examination	35 Marks
Continuous Internal Evaluation	15 Marks
Credits	1

Course Objectives: To help the students

1. Participate in group discussions and case studies with confidence and to make effective presentations. Also to learn the art of communication.
2. With- resume packaging, preparing and facing interviews.
3. Build an impressive personality through effective time management & goal setting, self-confidence and assertiveness.
4. Understand what constitutes proper grooming and etiquette in a professional environment. Also to understand academic ethics and value systems.
5. To understand the elements of research and hone their soft skills through a live, mini project.

Course Outcomes:

After successful completion of this course, student will be able to

1. Be effective communicators and participate in group discussions and case studies with confidence. Also be able to make presentations in a professional context.
2. Write resumes, prepare and face interviews confidently.
3. Be assertive and set short term and long term goals. Also learn to manage time effectively and deal with stress.
4. Make the transition smoothly from Campus to Corporate. Also use media with etiquette and know what academic ethics are.
5. To do a live, mini project by collecting and analyzing data and making oral and written presentation of the same.

Exercise 1

Group Discussion and Case studies: Dynamics of group discussion, intervention, summarizing, modulation of voice, body language, relevance, fluency and coherence.

Elements of effective presentation, Structure of presentation, Presentation tools, Body language, Creating an effective PPT.

Exercise 2

Interview Skills: Resume writing, structure and presentation, planning, defining the career objective, projecting ones strengths and skill-sets.

Interview Skills: concept and process, pre-interview planning, opening strategies, answering strategies, mock interviews.

Exercise 3

Personality Development: Effective Time Management, setting realistic goals, self-confidence and assertiveness, stress management, moral values.

Exercise 4

Corporate Culture: Grooming and etiquette, communication media etiquette, Academic ethics and integrity

Exercise 5

Mini Project: General/Technical Research, developing a questionnaire, data collection, analysis, written report and project seminar.

Suggested Reading:

1. Dr. Shalini Verma, “Body Language- Your Success Mantra”, S Chand, 2006.
2. Ramesh, Gopalswamy, and Mahadevan Ramesh, “The ACE of Soft Skills”, New Delhi: Pearson, 2010.
3. Covey and Stephen R, “The Habits of Highly Effective People”, New York: Free Press, 1989.

CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY (A)
Choice Based Credit System
B.E (Information Technology)

SEMESTER – IV

S.No.	Course Code	Title of the Course	Scheme of Instruction		Scheme of Examination			Credits
			Hours per week		Duration of SEE in Hours	Maximum Marks		
			L/T	P/D		CIE	SEE	
THEORY								
1	16IT C08	Design and Analysis of Algorithms	3/1	-	3	30	70	4
2	16IT C09	Data Communications	3	-	3	30	70	3
3	16IT C10	Java Programming	3	-	3	30	70	3
4	16IT C11	Computer Organization and Microprocessors	4	-	3	30	70	4
5	16IT C12	Electrical Machines and Microcontroller Applications	3	-	3	30	70	3
6	16MB C01	Engineering Economics and Accountancy	3	-	3	30	70	3
PRACTICALS								
7	16IT C13	Java Programming Lab	-	3	3	25	50	2
8	16IT C14	Microprocessors Lab	-	3	3	25	50	2
9	16IT C15	Mini Project - II	-	2	-	50	-	1
Total			**20**	**8**	**-**	**280**	**520**	**25**

L: Lecture T: Tutorial D: Drawing P: Practical

CIE - Continuous Internal Evaluation SEE - Semester End Examination

Assessment Procedures for Awarding Marks

The distribution of marks is based on CIE by concerned teacher and the Semester end examination shall be as follows:

Course (in terms of credits)	CIE	Semester end Examination(Marks)	Remarks	Duration of SemesterEnd Examination
Three(3) Credits/ Four(4) Credits	30*	70**	Theory Course/ Engg.Graphics	3 Hours
Two(2) Credits	20*	50***	Theory	2 Hours
Two(2) Credits	25	50	Lab Course/ Workshop	3 Hours
One(1) Credit	15	35	Lab Course	2 Hours
Two(2) Credits	50	—-	Project Seminar/ Seminar	——
Six(6) Credits	50	100	Project	Viva
One(1) Credit	—-	50***	Environmental Studies,Professional Ethics and Human values	2 Hours
One(1) Credit	50		Mini Project	——-

CIE: Continuous Internal Evaluation

*Out of 30/20 sessional marks(CIE), 10/5 marks are allotted for slip-tests (Three slips test will be conducted, each of 10/5 marks, best two average is considered) and the remaining 20/15 marks are based on the average of two tests, weightage for each test is 20/15 marks.

**The question paper will be in two parts, Part-A and Part-B. Part A is for Ten(10) questions and is compulsory, covers the entire syllabus, and carries 20 marks. Part-B carries 50 marks and covers all the units of the syllabus (student has to answer five out of seven questions).

***The question paper will be in two parts, Part-A and Part-B. Part A is for Ten(10) questions and is compulsory, covers the entire syllabus, and carries 15 marks. Part-B carries 35 marks and covers all the units of the syllabus (student has to answer five out of seven questions).

Note: A course that has CIE (sessional marks) but no semester end examination as per scheme, is treated as Pass/Fail for which pass marks are 50% of CIE.

A candidate has earned the credits of a particular course, if he/she secures not less than the minimum marks/grade as prescribed. Minimum pass marks for theory course is 40% of total marks i.e., CIE plus semester end examinations where as for the lab course/project is 50%.

16IT C08

DESIGN AND ANALYSIS OF ALGORITHMS

Instruction	3L+1T Hours per week
Duration of Semester End Examination	3 Hours
Semester End Examination	70 Marks
CIE	30 Marks
Credits	4

Course Objectives:

This course is introduced

1. To familiarize students with the concepts related to the design and analysis of algorithms.
2. To cover in detail greedy strategies, divide and conquer techniques, dynamic programming, back tracking and branch and bound for designing algorithms and illustrates them using a number of well-known problems and applications.
3. To describe two classes of problems NP hard and NP complete and discuss their applications.

Course outcomes:

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

1. Determine the class and the algorithm technique most suited to solve the problem in hand.
2. Compare between different data structures. Pick an appropriate data structure for a design situation.
3. Design algorithms of their own for different problems.
4. Synthesize/adapt an algorithm to solve the problem in hand and argue its correctness.
5. Analyse best-, average- and worst-case complexities of algorithms using asymptotic notations.
6. Identify the complexity classes such as P, NP, NP-Complete and NP-Hard to which an algorithm belongs and design a feasible solution.

Prerequisites:

Programming and Problem Solving (16CSC01), Discrete Structures (16ITC01), Data Structures (16ITC02).

UNIT-I

Introduction: Algorithm Specification, Performance analysis, Space Complexity, TimeComplexity, Asymptotic Notation (O, Omega, Theta), Practical Complexities, PerformanceMeasurement, Randomized Algorithms: An informal discussion, Review of elementary datastructures : Stacks, Queues, Trees, Heap and Heap Sort, Set representation, UNION, FIND.

UNIT-II

Divide- and Conquer: The general method, Finding the maximum minimum. Merge sort, Quick sort, Selection Problem, Strassen's Matrix Multiplication **Greedy Method:** The General Method,Knapsack problem, Job sequencing with deadlines, Minimum Cost Spanning Trees, Optimal Storage on tapes, Optimal merge patterns, Single Source Shortest Paths.

UNIT-III

Dynamic Programming: The General Method, Multistage graph, Single source shortest path, All Pair Shortest Path, Optimal Binary Search trees, 0/1 Knapsack, Reliability Design, Traveling Salesperson Problem, **Techniques for Graph Traversal**: Breadth First search and Traversal, Depth First Search and Traversal, Connected Components and Spanning Trees, Bi-connected Components .

UNIT-IV

Backtracking : The General Method, 8-Queens Problem, Graph Colouring, Hamilton cycle, Knapsack Problem, **Branch and Bounds:** The Method, LC Search, 15 puzzle, FIFO Branch and Bound, LC Branch and Bound, 0/1 Knapsack Problem, Traveling salesperson problem.

UNIT-V

NP-Hard and NP-Completeness: Basic concepts, Non-Deterministic Algorithms, The Classes NP Hard and NP Complete. Cook's theorem, NP-hard Graph Problems: Node Cover Decision Problem, Chromatic Number Decision Problem, Directed Hamiltonian Cycle, Traveling salesperson decision problem, NP Hard Scheduling Problems: Job Shop Scheduling.

Text Books:

1. Ellis Horowitz, Sartaj Sahani and Sanguthevar Rajasekaran, Fundamentals of Computer Algorithm, 2nd Edition, Universities Press, 2011.
2. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, "Introduction to Algorithms", 2nd Edition, Prentice Hall of India Private Limited, 2006.

Suggested Reading:

1. AnanyLevitin, "Introduction to the Design & Analysis of Algorithms", Pearson Education, 2003.
2. Aho, Hopcroft, Ullman, "The Design and Analysis of Computer Algorithm", Pearson Education, 2000.
3. Parag H.Dave, Himanshu B. Dave, "Design and Analysis of Algorithms", Pearson Education, Second Edition, 2014.

16IT C09

DATA COMMUNICATIONS

Instruction	3 Hours per week
Duration of Semester End Examination	3 Hours
Semester End Examination	70 Marks
CIE	30 Marks
Credits	3

Course Objectives:

This course is introduced to:

1. Familiarize students with the basics of data transmission, transmission media, data Communications System and its components.
2. Introduce various encoding, modulation schemes and data link protocols.
3. Give overview of different types of multiplexing and spread spectrum techniques.
4. Familiarize students with different types of Ethernet, architecture and services of WLANs and Bluetooth.

Course Outcomes:

After successful completion of this course, student will be able to

1. Demonstrate systematic understanding of Data Communication Techniques.
2. Apply various encoding schemes.
3. Understand multiplexing techniques.
4. Get acquainted with the concepts of virtual circuit networks.
5. Understand various types of switching techniques.
6. Understand concepts of wireless LANs.

Prerequisites:

Engineering physics(16PYC01), Applied physics(16PYC02).

UNIT-I

Data Communications, Data Networks and The Internet: Communications Model, Networks, The Internet, An Example Configuration, The Need for a Protocol Architecture, The TCP/IP Protocol Architecture, The OSI Model, Standardization within a Protocol.

Data Transmission: Concepts and Terminology, Analog and Digital Transmission, Transmission Impairments, Transmission media: Guided transmission media, wireless transmission.

UNIT-II

Data Encoding: Digital Data Digital Signals, Digital Data-Analog Signals, Analog Data-Digital Signals, Analog Data-Analog Signals.

Digital data communication techniques: Asynchronous and Synchronous Transmission, Types of errors, error detection, error correction, Line Configuration.

Data Link Control protocols: Flow Control, Error Control, HDLC**.**

UNIT-III

Multiplexing: Frequency Division Multiplexing,Wavelength Division Multiplexing, Synchronous Time Division Multiplexing, Statistical Time Division Multiplexing.

Spread Spectrum: The Concept, Frequency Hopping Spread Spectrum, Direct Sequence Spread Spectrum.

UNIT-IV

Circuit Switching and Packet Switching: Switched Communications Networks, Circuit-Switching Networks,Circuit-Switching Concepts, Soft switch Architecture,Packet-Switching Principles.

Virtual Circuit Networks: Frame Relay: Architecture, frame relay layers, extended address; ATM: Design goals, problems, architecture, switching, ATM layers.

UNIT-V

Traditional Ethernet: Topologies and Transmission Media, LAN protocol architecture, MAC sub layer, CSMA/CD, Physical Layer, Bridged, Switched and Full Duplex Ethernets Fast Ethernet: MAC sub Layer, Physical layer, Gigabit Ethernet: MAC sub Layer, Physical Layer.

Wireless LANs: Overview, Wireless LAN Technology, IEEE 802.11 Architecture and Services, IEEE 802.11 Medium Access Control, IEEE 802.11 Physical Layer. Bluetooth: Architecture, Layers.

Text Books:

1. Behrouz A. Forouzan, “Data Communications and Networking”, 4th edition, TataMcGraw Hill, 2006.
2. William Stallings, “Data and Computer communication”, 8th edition, Pearson Education, Asia-2004.

Suggested Reading:

1. Fred Halsall, “Data Communications, Computer Networks and Open Systems”, 4th edition, Pearson Education, 2000.
2. Andrew S. Tanenbaum, “Computer Networks”, 5th edition, Pearson Education.
3. Gilbert Held, “Understanding Data Communications”, 7th Edition, Pearson Education.

16IT C10

JAVA PROGRAMMING

Instruction	3 Hours per week
Duration of Semester End Examination	3 Hours
Semester End Examination	70 Marks
CIE	30 Marks
Credits	3

Prerequisites:
Programming and problem solving (16CSC01), Object Oriented Programming (16ITC303)

Course Objectives:

1. To introduce the fundamentals of object-oriented programming in Java which includes defining classes, invoking methods, inheritance, polymorphism, exception handling etc.
2. To familiarize students with event driven Graphical User Interface (GUI) programming and usage of standard class libraries.
3. To impart skills required to solve real world problems by creating Java applications using sound OOP practices, and APIs.

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

1. Achieve proficiency in object-oriented concepts and also learns to incorporate the same into the Java programming language.
2. Create Java application programs using sound OOP practices e.g. Inheritance, interfaces and proper program structuring by using packages, access control specifiers.
3. Understand and Implement the concepts of Exception Handling in java.
4. Develop the ability to solve real-world problems through software development in high-level programming language using Large APIs of Java as well as the Java standard class library.
5. Understand File, Streams, Input and Output Handling in java.
6. Create graphical user interface and Applets in java as well as apply the knowledge of Event Handling.

UNIT-I

Introduction to Java: Objects, Classes, Java Programs, Introduction to jdk and jre, Java Primitive Types, Basic Operators, Conditional and Logical statements, Some Typical Differences Between C and Java. **Defining Classes:** Adding Instance Fields and Methods, Constructors, Access Modifiers (Visibility Modes), Object Creation Examples, Method Overloading and Constructor Overloading, Use of static and final

keywords, Objects as parameters, Difference between local variable and instance field, Introduction to Object class, How to read user input from keyboard.

UNIT-II

Inheritance, Interfaces and Packages in Java: Defining super / sub classes, Abstract classes, Method overriding, Interfaces, Using Library Interfaces [Comparable &Comparator],**Packages:** Defining, Creating and Accessing a Package, Understanding CLASSPATH,importing packages, **Arrays, Strings in Java:** How to create and define arrays, Introduction to java.util.Array class, Difference between String &String Buffer classes, StringTokenizer class and Wrapper classes and conversion between Objects and primitives, **Inner classes in Java:** Types of inner classes, Creating static / non-static inner classes, Local and anonymous inner classes.

UNIT-III

Exception Handling in Java: What are exceptions, Error Vs. Exception, usage of try, catch, throw, throws and finally clauses, writing your own exception classes,Difference between checked Vs. unchecked Exceptions.**Generics:**What are Generics? Generic classes,bounded types, Generic methods and interfaces.

UNIT-IV

Collections: Overview of Java Collection Framework, Commonly used Collectionclasses – ArrayList, LinkedList, HashSet, TreeSet, HashMap, TreeMap, Collection Interfaces –Collection, Set, List, Map, Legacy Collection classes – Vector, Hashtable, Stack,Enumeration interface, Iteration over Collections – Iterator and ListIterator interfaces. **File Handling:** Stream classes, Reader and Writer classes, File and Directory class.

UNIT-V

GUI Design & Event Handling: Component, Container, Color,GUI Controls, Layout Managers, Introduction to Swings, Delegation Event Model, Event Classes, Source of Events, Event Listener Interfaces. Handling button click, mouse and keyboard events, and Adapter classes. Writing GUI Based applications, Applets, life cycle of an Applet, Developing and running applets, passing parametersto applets. **Database Handling in Java:** Java Database Connectivity (JDBC).

Text Books:

1. Herbert Schildt: "Java: The Complete Reference", 8th Edition, Tata McGraw Hill Publications, 2011.
2. Cay S. Horstmann, Gary Cornell: "Core Java, Volume I—Fundamentals", 8th edition, Prentice Hall, 2008.

Suggested Reading:

1. Sachin Malhotra&SaurabhChoudhary: "Programming in Java", 2nd Edition, Oxford University Press, 2014.
2. K. Arnold and J. Gosling, "The JAVA programming language", 3rd Edition, Pearson Education, 2000.
3. Timothy Budd, "Understanding Object-oriented programming with Java", Addison-Wesley, 2002.
4. C. Thomas Wu, "An introduction to Object-oriented programming with Java", 4th edition, Tata McGraw-Hill Publishing company Ltd., 2010.
5. James M Slack, Programming and Problem Solving with JAVA, Thomson Learning, 2002.

16IT C11

COMPUTER ORGANIZATION AND MICROPROCESSORS

Instruction	4 Hours per week
Duration of Semester End Examination	3 Hours
Semester End Examination	70 Marks
CIE	30 Marks
Credits	4

Course Objectives:

1. To familiarize students with the design and organization of a digital computer, operation of various functional units, instruction set design and factors that influence the performance of a computer.
2. To present students with concepts of memory system and its types.
3. To facilitate students with the understanding of architecture and instruction set of 8085 in particular and programming 8085.
4. To facilitate students with the understanding of the functionality and interfacing of various peripheral devices.

Course Outcomes:

After successful completion of this course, student will be able to

1. Understand and analyze the performance of computer systems and know how to improve their efficiency.
2. Get acquainted with the concepts of computer Arithmetic operations.
3. Understand the internal organization of memory system and various types of memory unit.
4. Understand the architecture and instruction set of 8085.
5. Write assembly language programs using 8085 instruction set.
6. Understand interfacing with various peripheral devices.

Prerequisites:

Digital Electronics and Logic Design (16ITC04).

UNIT I

Basic Structure of Computers: Computer Types, Functional Units, Basic Operational Concepts, Bus Structures, Software, Performance, Multiprocessors and Multi-computers, Historical perspective. **Arithmetic:** Addition and Subtraction of Signed numbers: Addition/Subtraction Logic Unit, Design of fast adders: Carry – Look-ahead Addition, Multiplication of positive numbers, Signed-Operand Multiplication: Booth Algorithm, Fast Multiplication: Bit-Pair Recording of Multipliers, Carry-Save addition of Summands, Integer Division, Floating Point Numbers and Operations: IEEE Standard for Floating-Point Numbers, Arithmetic Operations on

Floating-Point Numbers, Guard Bits and Truncation, Implementing Floating-Point Operations.

UNIT II

The Memory System: Basic concepts, Semi-conductor RAM Memories: Internal Organization of Memory Chips, Static Memories, Asynchronous DRAMs, Synchronous DRAMs , Structure of Larger Memories, Memory System Considerations, Rambus Memory, Read-Only Memories: ROM, PROM, EPROM, EEPROM, Flash Memory, Speed, Size and Cost, Cache Memories**:** Mapping Functions, Replacement Algorithms, Performance considerations: Interleaving, Hit rate and Miss Penalty, Caches on the Processor Chip, Other Enhancements. Virtual Memories: Address Translation, Memory Management requirements, Secondary Storage: Magnetic Hard Disks, Optical Disks and Magnetic Tape Systems.

UNIT III

8085 Microprocessor Architecture: Introduction to Microprocessors, The 8085 MPU: The 8085 Microprocessor, Microprocessor Communication and Bus Timings, De-multiplexing the Bus AD7-AD0, Generating Control Signals, A Detailed Look at the 8085 MPU and its Architecture, Decoding and Executing an Instruction.

Programming the 8085: Introduction to 8085 instructions: Data Transfer Operations, Arithmetic Operation, Logic Operations, Branch Operations, Writing Assembly Language Programs, Debugging a Program. Programming techniques with Additional instructions: Programming Techniques-Looping, Counting and Indexing, Additional Data Transfer and 16-Bit Arithmetic Instructions, Arithmetic Operations Related to memory, Logic Operations: Rotate and Compare, Dynamic Debugging.

UNIT IV

Stacks and subroutines: Stack, Subroutine, Restart, Conditional CALL and RETURN instructions, Advanced Subroutine Concepts.

Interrupts: The 8085 Interrupt, 8085 Vectored Interrupts: TRAP, RST 7.5, 6.5, AND 5.5, Additional I/O Concepts and Processes: Programmable Interrupt Controller (8259A), Direct Memory Access (DMA) and 8257 DMA controller.

Interfacing Data Converters: Digital to Analog (D/A) Converters, Analog to Digital (A/D) Converters.

UNIT V

Programmable Peripheral Interface (Intel 8255A), Programmable Communication Interface (Intel 8251), Programmable Interval Timer (Intel 8253 and 8254), Programmable Keyboard /Display Controller (Intel 8279),

Serial and Parallel bus Standards: RS 232 C and IEEE 488. Hardware controlled serial I/O using programmable chips: 8251.

Text Books:

1. Carl Hamacher, ZvonkoVranesic, SafwatZaky, "Computer Organization", 5th Edition, McGraw Hill, 2002.
2. Ramesh S Gaonkar, "Microprocessor Architecture, Programming and Applications with the8085", 5th edition, Prentice Hall, 2002.

Suggested Reading:

1. M. M. Mano, "Computer System Architecture", 3rd edition, Prentice Hall, 1994.
2. William Stallings, "Computer Organisation and Architecture, Design for Performance", Pearson, 9th Edition, 2013.
3. Pal Chouduri, "Computer Organization and Design", Prentice Hall of India, 1994.
4. Douglass V. Hall,"Microprocessors and Interfacing: Programming and Hardware", 2nd Edition.

16IT C12

FUNDAMENTALS OF DATA SCIENCE

Instruction	3 Hours per week
Duration of Semester End Examination	3 Hours
Semester End Examination	70 Marks
CIE	30 Marks
Credits	3

Course Objectives:

1. To introduce statistical tools for analyzing empirical data.
2. To Familiarize students with different types of regression models.
3. To introduce various types of tests for comparison of means and variances of different datasets.
4. To familiarize students with data structures in R and packages for effective representation of results.

Course Outcomes:

After successful completion of this course, student will be able to

1. Analyze data using Classification, Graphical and computational methods.
2. Apply statistical methods to data for inferences.
3. Perform descriptive analytics over massive data.
4. Create or read from external datasets.
5. Store, retrieve and manipulate using R data structures.
6. Perform data manipulation, statistical analysis and present their results in well-formatted textual and graphical formats.

UNIT-I

Descriptive statistics: Measure of Central Tendency (Mean, Median and Mode), Measure of Positions (Quartiles, Deciles, Percentiles and Quantiles), Measure of Dispersion (Range, Median, Absolute deviation about median, Variance and Standard deviation), Measure of Distribution (Skewness and Kurtosis), Dependent independent events and Bayesian Approach, Graphical Analysis: Histograms and frequency polygons, Box-plots-Box and Whisker Plot-Box Plot and its parts, Using Box Plots to compare distribution,Scatter Plots, Heat Maps.

UNIT-II

Random variables and Regression Models: Overview of Random variables and distributions, Mathematical expectation, variance and covariance, Linear Regression: Correlation coefficient, Simple, multiple and logistic regressions, LDA and comparison of classification methods.

UNIT-III

Inferential statistics: Comparing Population-Test of Hypothesis (Concept of Hypothesis testing, Null Hypothesis and Alternative Hypothesis), Cross

Tabulations (Contingency table and their use, Chi-Square test, F-test), One Sample t-test (Concept, Assumptions, Hypothesis, Verification of assumptions, Performing the test and interpretation of results), Independent Samples t-test (Concept, Type, Assumptions, Hypothesis, Verification of assumptions, Performing the test and interpretation of results), Paired Samples t-test (Concept, Assumptions, Hypothesis, Verification of assumptions, Performing the test and interpretation of results), One way ANOVA (Concept, Assumptions, Hypothesis, Verification of assumptions, Model fit, Hypothesis testing).

UNIT-IV

Introduction to R: Installing R in windows, R Console (R window to edit and execute R Commands), Commands and Syntax (R commands and R syntax), Packages and Libraries (Install and load a package in R), Help In R, Workspace in R.

Familiarity of Data Structures in R: Introduction to Data Types (Why Data Structures?, Types of Data Structures in R), Vectors, Matrices, Arrays, Lists, Factors, Data Frames, Importing and Exporting Data.

UNIT-V

Graphical Analysis using R: Creating a simple graph (Using plot() command), Modifying the points and lines of a graph (Using type, pch, font, cex, lty, lwd, col arguments in plot() command), Modifying Title and Subtitle of graph (Using main, sub, col.main, col.sub, cex.main, cex.sub, font.main, font.sub arguments in plot() command), Modifying Axes of a Graph (Using xlab, ylab, col.lab, cex.lab, font.lab, xlim, ylim, col.axis, cex.axis, font.axis arguments and axis() command), Adding Additional Elements to a Graph (Using points(), text(), abline(), curve() commands), Adding Legend on a Graph (Using legend() command), Special Graphs (Using pie(), barplot(), hist() commands), Multiple Plots (Using mfrow or mfcol arguments in par() command and layout command).

Text Books:

1. Fundamentals of Mathematical statistics by S.C. Gupta and V.K. Kapur, Eleventh Edition Sultan Chand &Sons publications (Reprint) 2014. (Units 1,2 and 3).
2. An Introduction to Statistical Learning with Applications in R, Gareth James Daniela Witten Trevor Hastie, Robert Tibshirani, February 11, 2013, web link: www.statlearning.com. (Units 4 and 5).

Suggested Reading:

1. Beginning R The statistical Programming Language, Mark Gardener , Wiley, 2015.
2. Data Science and Big Data Analytics, EMC Education Services, EMC^2, Wiley Publication, 2015.

16MB C01

ENGINEERING ECONOMICS AND ACCOUNTANCY

Instruction	3 hours per week
Duration of Semester End Examination	3 Hours
Semester End Examination	70 Marks
Continuous Internal Evaluation	30 Marks
Credits	3

Course Objectives:

The Objectives of the course are

1. to introduce managerial economics and demonstrate its importance in managerial decision making.
2. to develop an understanding of demand and relevance of its forecasting in the business.
3. to provide the basics of market structure and the concept of equilibrium in different market structures.
4. to examine the economic analysis of production process, types of inputs and to explain different costs and their relationship with the output.
5. to understand the importance of project evaluation in achieving a firm's objective.
6. to explain the concept of Accountancy and provided knowledge on preparation and analysis of Final accounts.

Course Outcomes:

After successful completion of this course, student will be able to

1. Apply fundamental knowledge of Managerial economics concepts and tools.
2. Understand various aspects of demand analysis and forecasting.
3. Understand price determination for different markets.
4. Study production theory and analyze various costs & benefits involved in it so as to make best use of resources available.
5. Analyze different opportunities and come out with best feasible capital investment decisions.
6. Apply accountancy concepts and conventions, Final accounts and financial analysis.

UNIT-I:

Introduction to Managerial Economics

Introduction to Economics and its evolution - Managerial Economics - its scope, importance, Its usefulness to engineers - Basic concepts of Managerial economics.

UNIT-II:

Demand Analysis

Demand Analysis - Concept of demand, determinants, Law of demand, its assumptions, Elasticity of demand, price, income and cross elasticity, Demand Forecasting – Types of Market structures. (Simple numerical problems).

UNIT-III:

Production and Cost Analysis

Theory of Production - Firm and Industry - Production function - input-out relations - laws of returns - internal and external economies of scale. Cost Analysis: Cost concepts - fixed and variable costs - explicit and implicit costs - out of pocket costs and imputed costs - Opportunity cost - Cost output relationship - Break-even analysis. (Theory and problems).

UNIT-IV:

Accountancy

Book-keeping, principles and significance of double entry book keeping, Journal, Subsidiary books, Ledger accounts, Trial Balance concept and preparation of Final Accounts with simple adjustments.

UNIT-V:

Capital Budgeting

Introduction to capital budgeting, Methods: traditional and discounted cash flow methods. Introduction to Working capital management. (Numerical problems).

Text Books:

1. Mehta P.L., "Managerial Economics – Analysis, Problems and Cases", Sultan Chand & Son's Educational publishers, 2013.
2. Maheswari S.N. "Introduction to Accountancy", Vikas Publishing House, 2013.
3. Panday I.M. "Financial Management", Vikas Publishing House, 11th edition, 2015.

Suggested Readings:

1. Varshney and KL Maheswari, Managerial Economics, Sultan Chand, 2014.
2. M.Kasi Reddy and S.Saraswathi, Managerial Economics and Financial Accounting, Prentice Hall of India Pvt Ltd, 2007.
3. A.R.Aryasri, Managerial Economics and Financial Analysis, McGraw-Hill, 2013.

16IT C13

JAVA PROGRAMMING LAB

Instruction	3 Hours per week
Duration of Semester End Examination	3 Hours
Semester End Examination	50 Marks
CIE	25 Marks
Credits	2

Course Objectives:

1. Be able to use the Java SDK environment to create, debug and run simple Java programs.
2. To build development skills using java programming for real world applications.
3. To implement frontend and backend of java based applications.

Course Outcomes:

After successful completion of this course, student will be able to

1. Develop Java applications using the concepts of Inheritance, interfaces, packages, access control specifiers.
2. Implement the concepts of Exception Handling in java Applications.
3. Read and write data using different Java I/O streams.
4. Create graphical user interfaces and Applets by applying the knowledge of Event Handling.
5. Create robust applications using Java standard class libraries and retrieve data from a database with JDBC.
6. Ability to solve real-world problems by designing user friendly GUI with befitting backend through the APIs of Java.

List of programs

1. Program(s) to illustrate the concepts of constructor overloading, method overloading, static and final keywords usage.
2. Program(s) to illustrate the concepts of Inheritance, method overriding, super keyword usage, Dynamic polymorphism.
3. Program(s) to illustrate concept of abstract class & interfaces, Comparator and Comparable interfaces.
4. Program(s) to demonstrate String handling with String, StringBuffer and StringTokenizer classes.
5. Program(s) to demonstrate various types of inner classes, Packages creation and usage.
6. Program(s) to demonstrate concept of exception handling and user defined exceptions.
7. Program(s) using Generics, Collection framework classes and Interfaces.
8. Program(s) to illustrate the usage of I/O streams.

9. Program(s) to illustrate GUI with different controls, event handling and applets.
10. Program to connect to a database using JDBC.

Suggested Reading:

1. Herbert Schildt: "Java™: The Complete Reference Java", 8th edition, Tata McGraw Hill Publications, 2011.
2. Sachin Malhotra&SaurabhChoudhary: "Programming in Java", 2nd Edition, Oxford University Press, 2014.

16IT C14

MICROPROCESSORS LAB

Instruction	3 Hours per week
Duration of Semester End Examination	3 Hours
Semester End Examination	50 Marks
CIE	25 Marks
Credits	2

Course Objectives:

1. To familiarize students with the architecture and Instruction set of Intel 8085 microprocessor.
2. To provide practical hands on experience with Assembly Language Programming.
3. To impart skills required to interface various peripheral devices with 8085 microprocessor.

Course Outcomes:

After successful completion of this course, student will be able to

1. Describe the architecture and comprehend the instruction set of 8085.
2. Understand and apply the principles of Assembly Language Programming in developing microprocessor based applications.
3. Write assembly language programs using Arithmetic and logic instructions.
4. Write assembly language programs using branch and conditional instructions.
5. Write assembly language programs using stacks and sub routines.
6. Work with standard microprocessor interfaces like stepper motor, digital-to-analog Converters and analog-to-digital converters etc.

Prerequisites:

Digital Electronics and Logic Design (16ITC04).

List of Experiments

1. Introduction to 8085 instruction setand microprocessor trainer kit.
2. Assembly language programs using Arithmetic and logic instructions.
3. Assembly language programs using branch and conditional instructions.
4. Assembly language programs using stacks and sub routines.
5. Interfacing and programming of 8255. (E.g. traffic light controller).
6. A/D converter interface.
7. D/A converter interface.
8. Speed and Direction Control of Stepper Motor.
9. Practice Programs using 8085 Simulator.

Text Books:

1. Ramesh S Gaonkar, “Microprocessor Architecture, Programming and Applications withthe 8085”, 5th edition, Prentice Hall, 2002.

16IT C15

MINI PROJECT – II

Instruction	2 Hours per week
Duration of Semester End Examination	-
Semester End Examination	-
CIE	50 Marks
Credits	1

Course Objectives:

1. To learn by doing, by taking responsibility of the end product.
2. To develop capability to analyse and solve real world problems with an emphasis on applying/integrating knowledge acquired.

Course Outcomes:

After successful completion of this course, student will be able to

1. Construct innovative solutions.
2. To work in team as well as individuals.
3. To manage time and resources.

The Students are required to implement one of the projects from project exercise given in the suggested readings of the theory subjects. Focus may be on File structures, Micro Processor Based Projects. During the implementation of the project, Personnel Software Process (PSP) has to be followed. Report of the project work has to be submitted for evaluation.