Learning Outcomes based Curriculum Framework (LOCF)

For

Bachelor of Technology
Computer Science & Engineering
Four-Year Graduate Programme

Curriculum for 2nd Year

2022-23 Onward

Faculty of Engineering and Technology
Chaudhary Devi Lal University
Sirsa-125055

Program Specific Outcomes (PSOs)

PSO1 **Developing Computational Systems**: Use principles of electronics and Micro-Processors, various programming languages, data structures, database management systems, computer algorithms, theory of computation and software engineering for designing and implementing computational systems.

PSO2 **Devising Networking Solutions**: Apply the knowledge of systems in the areas related to network technologies, mobile ad hoc and sensor networks, cloud computing, IoT and, information and web security for devising networking solutions.

PSO3 **Doing Data Analytics and Designing Intelligent Systems**: Utilize the approaches and tools of artificial intelligence and soft computing, data analytics and machine learning for designing and working with intelligent systems that can extract valuable information from large amount of data and learn from their environment.

Programme Outcomes (POs) of Bachelor Programmes in Engineering & Technology have been specified in First Year common curriculum of B.Tech. Programmes.

Course Code	**Definition/ Category**
BSC	Basic Science Courses
ESC	Engineering Science Courses
HSMC	Humanities and Social Sciences including Management Courses
MC	Mandatory Audit Courses
PC	Program Core
PE	Program Elective Courses
OE	Open Elective Courses
EEC	Employability Enhancement Courses (Project work/ Summer Training/ Industrial Training/ Practical Training/ Internship/Seminar, etc.)

Courses codes, titles, and credits (Semester- III)

#	Course Code	Course Title	Workload/Credit			
			Theory	Tutorial	Practical	Total
1	BSC/7-T	Mathematics-III	3/3	-/-	-	3/3
2	PC/CSE/1-T	Data Structures and Algorithms	3/3	-/-	-	3/3
3	PC/CSE/2-T	Object Oriented Programming using C++	3/3	-/-	-	3/3
4	PC/CSE/3-T	Discrete Mathematics	3/3	-/-	-	3/3
5	PC/CSE/4-T	Computer Organisation and Architecture	3/3	-/-	-	3/3
6	*MC/2-T	Environmental Science	3/-	-/-	-	3/-
7	PC/CSE/1-P	Data Structures and Algorithms using C/C++ Lab.	-/-	-/-	4/2	4/2
8	PC/CSE/2-P	Object Oriented Programming using C++ Lab.	-/-	-/-	4/2	4/2
		Total	**18/15**	**-/-**	**8/4**	**26/19**

Courses' codes, titles, and credits (Semester- IV)

#	Course Code	Course Title	Workload/Credit			
			Theory	Tutorial	Practical	Total
1	PC/CSE/5-T	Microprocessor and Interfacing	3/3	-/-	-	3/3
2	PC/CSE/6-T	Computer Networks	3/3	-/-	-	3/3
3	PC/CSE/7-T	Database Management System	3/3	-/-	-	3/3
4	PC/CSE/8-T	Analysis and Design of Algorithms	3/3	-/-	-	3/3
5	PC/CSE/9-T	Software Engineering	3/3	-/-	-	3/3
6	PC/CSE/10-T	Java Programming	3/3	-/-	-	3/3
7	PC/CSE/5-P	Microprocessor and Interfacing Lab.	-/-	-/-	2/1	2/1
8	PC/CSE/6-P	Computer Networks Lab.	-/-	-/-	2/1	2/1
9	PC/CSE/7-P	Database Management System Lab.	-/-	-/-	2/1	2/1
10	PC/CSE/10-P	Java Programming Lab.	-/-	-/-	4/2	4/2
		Total	**18/18**	**-/-**	**10/5**	**28/23**
1.	**EEC/CSE/1	Industrial Training/ Internship	-	-	-/4	-/4

*Non-credit qualifying mandatory course.

**The students will have to undergo Industrial/Practical Training/ Internship for 4-6 weeks during summer vacations after the examination of 4th semester which will be evaluated in 5th semester.

Note: Students will be allowed to use non-programmable scientific calculators only, however, sharing of calculator should not be permitted.

Detailed Syllabus of

B.Tech.(CSE)

3rd Semester

Mathematics III

General Course Information

Course Code: BSC/7-T Course Credits: 3 Type: Basic Sciences Contact Hours: 3 hours/week Mode: Lectures (L) Examination Duration: 3 hours	**Course Assessment Methods (internal: 30; external: 70)** Two minor examinations (20 marks), Class Performance measured through percentage of lectures attended (4 marks), assignments (6 marks), and the end- semester examination (70 marks). For the end semester examination, nine questions are to be set by the examiner. A candidate is required to attempt 5 questions in all. All questions carry equal marks. Question number 1 will be compulsory and based on the entire syllabus. It will contain seven parts of 2 marks each. Question numbers 2 to 9 will be given by setting two questions from each of the four units of the syllabus. A candidate is required to attempt the remaining four questions by selecting one question from each of the four units.

Pre-requisites: Mathematics I and Mathematics II

About the Course

This is an advanced mathematics course that offers the knowledge of Fourier Series, Fourier Transforms, Functions of Complex Variables. These concepts are essential for students to solve problems in image processing, digital signal processing and other related engineering fields.

Course Outcomes: By the end of the course students will be able to:

CO1. **define** concepts and terminology of Fourier Series and Fourier transforms, Functions of complex variables and Power Series etc. (LOTS: Level 1: Remember)

CO2. **solve** problems using Fourier transforms in domains like digital electronics and image processing. (LOTS: Level 3: Apply)

CO3. **apply** principles of functions of complex variables to solve computational problems. (LOTS: Level 3: Apply)

CO4. **compare** various concepts related to Fourier transforms and functions of complex variables. (HOTS: Level 4: Analyse)

CO5. **select** suitable method for given computational engineering problems and related domain. (HOTS: Level 4: Evaluate)

CO6. **integrate** the knowledge of Fourier Series and Fourier transforms, Functions of complex variables, and Power Series for solving real world problems. (HOTS: Level 6: Create)

Course Content

Unit I

Fourier Series and Fourier Transforms: Euler's formulae, conditions for a Fourier expansion, change of interval, Fourier expansion of odd and even functions, Fourier expansion of square wave, rectangular wave, saw-toothed wave, half and full rectified wave, half range sine and cosine series.

Unit II

Fourier integrals, Fourier transforms, shifting theorem (both on time and frequency axes), Fourier transforms of derivatives, Fourier transforms of integrals, Convolution theorem, Fourier transform of Dirac delta function.
Linear Programming Problem (LPP): Introduction; Formulation of linear programming problem (LPP); Graphical method for its solution; Standard form of LPP; Basic feasible solutions; Simplex Method and Dual Simplex Method for solving LPP.

Unit III

Functions of Complex Variable: Definition, Exponential function, Trigonometric and Hyperbolic functions, Logarithmic functions. Limit and Continuity of a function, Differentiability and Analyticity. Cauchy-Riemann equations, necessary and sufficient conditions for a function to be analytic, polar form of the Cauchy-Riemann equations. Harmonic functions.

Unit IV

Complex integral, Cauchy Gaursat theorem (without proof), Cauchy integral formula (without proof), Power series, radius and circle of convergence, Taylor's Maclaurin's and Laurent's series. Zeroes and singularities of complex functions, Residues. Evaluation of real integrals using residues (around unit and semi-circle only).

T**ext and Reference Books:**

1. F. Kreyszig, *Advanced Engineering Mathematics*, 10th edition, Wiley, 2015.
2. B. S. Grewal, *Higher Engineering Mathematics*, Khanna Publishers, 44th edition, 1965.
3. R. K. Jain, S.R.K. Iyenger. *Advance Engineering. Mathematics*, 4th edition, Narosa Publishing House, 2012.
4. Michael D. Greenberg, *Advanced Engineering Mathematics*, 2nd edition, Pearson Education, 2002.
5. Johnson and Miller *Probability and statistics for Engineers*, 8th edition, Pearson Education India, 2015.

CO-PO Articulation Matrix Mathematics-III (BSC/7-T)

List of Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1. **Define** concepts and terminology of Fourier Series and Fourier transforms, Functions of complex variables and Power Series. (LOTS: Level 1: Remember)	**1**	**--**	**-**		**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**2**	**2**	**2**
CO2. **Solve** problems using Fourier transforms in domains like digital electronics and image processing. (LOTS: Level 3: Apply)	**2**	**2**	**2**	**2**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**3**	**2**	**2**
CO3. **Apply** principles of functions of complex variables to solve computational problems. (LOTS: Level 3: Apply). (LOTS: Level 3: Apply)	**2**	**2**	**2**	**2**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**3**	**2**	**3**
CO4. **Compare** various concepts related to Fourier transforms and functions of complex variables (HOTS: Level 4: Analyse).	**3**	**3**	**2**	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**3**	**2**	**3**
CO5. **Select** suitable method for given computational engineering problems and related domain. (HOTS: Level 4: Evaluate)	**3**	**3**	**2**	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**3**	**2**	**3**
CO6. **Integrate** the knowledge of Fourier Series and Fourier transforms, Functions of complex variables and Power Series for solving real world problems. (HOTS: Level 6: Create)	**3**	**3**	**2**	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**2**	**2**	**3**
Level of Attainments BSC/7-T															

Data Structures and Algorithms

General Course Information

Course Code: PC/CSE/1-T Course Credits: 3 Type: Professional Core Contact Hours: 3hours/week Mode: Lectures (L) Examination Duration: 3 hours	**Course Assessment Methods (internal: 30; external: 70)** Two minor examinations (20 marks), Class Performance measured through percentage of lectures attended (4 marks), assignments (6 marks), and the end-semester examination (70 marks). For the end semester examination, nine questions are to be set by the examiner. A candidate is required to attempt 5 questions in all. All questions carry equal marks. Question number 1 will be compulsory and based on the entire syllabus. It will contain seven parts of 2 marks each. Question numbers 2 to 9 will be given by setting two questions from each of the four units of the syllabus. A candidate is required to attempt the remaining four questions by selecting one question from each of the four units.

Pre-requisites: Programming in C

About the Course:
Data Structure and Algorithms is a core and an essential course for every graduate in Computer Science and Engineering. This course introduces data structures like arrays, linked lists, trees and graphs etc. and various operations to be implemented on these data structures for solving real world problems. It includes various sorting and searching algorithms as well. Further, it incorporates complexity analysis of algorithms implemented on various data structures.

Course Outcomes: By the end of the course students will be able to:

CO1. **describe** various types of data structures and operations that can be implemented on these data structures. (LOTS: Level 1: Remember)
CO2. **demonstrate** the use of various data structures and their related operations. (LOTS: Level 2: Understand)
CO3. **apply** data structure to solve computational problems. (LOTS: Level 3: Apply)
CO4. **compare** the suitability of alternative data structures and prescribed operations for various problem situations. (HOTS: Level 4: Analyse).
CO5. **defend** solutions with respect to effective storage of data and efficiency of the required operations for solving real world problems. (HOTS: Level 5: Evaluate)

Course Content

Unit I

Introduction to data structures and their types, Abstract data types, Linear lists: Arrays and linked lists: memory representations, implementing operations like traversing, searching, inserting and deleting etc. Applications of arrays and linked lists. Representing sets and polynomials using linked lists.

Unit II

Stack and Queue: Static and linked implementations, Operations and Applications. Circular queues, Tress, Binary trees and related terminology, Tree traversals (Recursive), Threaded Binary Trees, Binary Search Trees implementation and operations, Priority queues.

Unit III

Height Balanced or AVL trees and B trees. Graph definitions and related terminology, memory representations and related operations (traversal, insertion, deletion, search), Path Matrix, Warshall's Shortest path algorithm Hashing, Hash tables, hash function and collision resolution.

Unit IV

Sequential and binary search, Sorting algorithms: Bubble sort, Selection sort, Insertion sort, Quick sort, Merge sort, Count sort, Heap sort, Comparison of searching and sorting techniques based on their complexity analysis, Time and space complexity of algorithms: Asymptotic analysis, Big O, Omega, Theta notations.

T**ext and Reference Books:**

1. Aho, A. V., Ullman, J. D., and Hopcroft, J. E., *Data Structures and Algorithms*, Addison-Wesley, 1983.
2. LangsamYedidyah, Augenstein J Moshe, Tenenbaum M Aaron, *Data Structures using C and C++*, 3rdedition, PHI, 2009.
3. Cormen, T. H., Leiserson, C. E., Rivest, R. L. and Stein, C., *Introduction to Algorithms*, MIT Press, 2009.
4. Robert L. Kruse, *Data Structure and Program Design in C*, Pearson Education India, 2007.
5. Weiss, M. A., *Data Structures and Algorithm Analysis in C++*, Addison-Wesley, 2007.
6. Sahni, S., *Data Structures, Algorithms, and Applications in C++*, WCB/McGraw-Hill, 2001.

CO-PO Articulation Matrix Data Structures and Algorithms Course (PC/CSE/1-T)

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1. **Describe** various types of data structures and operations that can be implemented on these data structures. (LOTS: Level 1: Remember)	1	-	-	-	-	-	-	-	-	-	-	-	3	-	-
CO2. **Demonstrate** the use of various data structures and their related operations. (LOTS: Level 2: Understand)	1	-	-	-	-	-	-	-	-	-	-	-	3	-	-
CO3. **Apply** data structure to solve computational problems. (LOTS: Level 3: Apply)	2	2	-	-	2	-	-	-	-	-	-	-	3	2	2
CO4. **Compare** the suitability of alternative data structures and prescribed operations for solving a problem. (HOTS: Level 4: Analyse).	2	2	-	-	-	-	-		-	-	-	-	3	2	2
CO5. **Defend** solutions with respect to effective storage of data and efficiency of the required operations for solving computational problems. (HOTS: Level 5: -Evaluate)	3	3	-	1	-	-	-	-	-	-	-	-	3	2	2
Level of Attainments PC/CSE/1-T															

Object Oriented Programming using C++

General Course Information

Course Code: PC/CSE/2-T Course Credits: 3 Type: Professional Core Contact Hours: 3 hours/week Mode: Lectures (L) Examination Duration: 3 hours	**Course Assessment Methods (internal: 30; external: 70)** Two minor examinations (20 marks), Class Performance measured through percentage of lectures attended (4 marks), assignments (6 marks), and the end- semester examination (70 marks). For the end semester examination, nine questions are to be set by the examiner. A candidate is required to attempt 5 questions in all. All questions carry equal marks. Question number 1 will be compulsory and based on the entire syllabus. It will contain seven parts of 2 marks each. Question numbers 2 to 9 will be given by setting two questions from each of the four units of the syllabus. A candidate is required to attempt the remaining four questions by selecting one question from each of the four units.

Pre-requisites: Knowledge of computer fundamentals and problem-solving using C programming

About the Course:
Objected Oriented Programming using C++ is an essential course for every graduate in Computer Science and Engineering. This course introduces the Object-Oriented concepts such as data encapsulation, data hiding, data abstraction, reusability, exception handling etc., and their implementation using C++.

Course Outcomes: By the end of the course students will be able to:

CO1. **List** the concepts related to object-oriented paradigms. (LOTS: Level 1: Remember)
CO2. **Distinguish** between structured and object-oriented approaches to programming. (LOTS: Level 2: Understand)
CO3. **Apply** object-oriented constructs for problem solving. (LOTS: Level 3: Apply)
CO4. **Detect** logical and run time errors and suggest appropriate modifications. (HOTS: Level 4: Analyze)
CO5. **Justify** the design of a program for a given problem. (HOTS: Level 5: Evaluate)
CO6. **Design** solutions to programming problems using multiple object-oriented programming constructs together. (HOTS: Level 6: Create)

Course Content

Unit I

Introduction to object-oriented programming, C++ standard library, basics of a typical C++ environment, illustrative simple C++ programs, new features of ANSI C++ standard, OOPs concepts: Information hiding, encapsulation, data abstraction, access modifiers, controlling access to a class level, method, or variable (public, protected, private, block level, scope and mutable), other modifiers. Structure of class and struct in memory, accessing members of structures, Class scope and accessing class members, separating interface from implementation, pre-processors directives, macro programs, header files and namespaces, default constructors, chained constructor, default arguments with constructors, constant object and const member

functions, object as member of class, use of destructors, virtual destructors, controlling access function and utility functions, function overloading.

Unit II

Inline function, friend function and friend classes, using this pointer, dynamic memory allocation with new and delete, static class members, proxy class, polymorphism concepts, overloading, overriding methods, abstract classes, reusability, class's behaviors, inheritance, base classes and derived classes, protected members, casting base-class pointers to derived-class pointers, using member functions, overriding base-class members in a derived-class, public, protected and private inheritance, using constructors and destructors in derived classes, implicit derived-class object to base- class object conversion, composition vs. inheritance.

Unit III

Virtual functions, abstract base classes and concrete classes, new classes and dynamic binding, virtual destructors, fundamentals of operator overloading, restrictions on operators overloading, operator functions as class members vs. as friend functions, overloading, <<, >> overloading unary operators, overloading binary operators. I/O Streams, files handling, creating a sequential access file, reading data from a sequential access file, updating sequential access files, random access files, creating a random-access file, writing data randomly to a random-access file, reading data sequentially from a random-access file.

Unit IV

Managing Console I/O, stream input/output classes and objects, stream output, stream input, unformatted I/O (with read and write), stream manipulators, stream format states, stream error states, exception handling, basics of C++ exception handling(try, throw, catch), rethrowing an exception, specific exception, processing unexpected exceptions, stack unwinding, exception handling in constructors and destructors, inheritance with exception introduction to generic classes, function templates, overloading template functions, class template, non-type parameters, templates and inheritance, templates and friends, templates and static members, container, iterator, algorithm and functional classes.

T**ext and Reference Books:**

1. H. M. Deitel and P. J. Deitel, *C++ How To Program*, 6th Ed., Prentice Hall, 2008.
2. Robert Lafore, *Object-Oriented Programming in C++*, 3rd Ed., Sams Publishing, 2001.
3. D. Ravichandran, *Programming with C++*, 3rd Ed., T.M.H, 2011.
4. E. Balagurusamy, *Object oriented Programming with C++*, 6th Ed., Tata McGraw-Hill, 2013.
5. Horstmann, *Computing Concepts with C++ Essentials*, 3rd Ed., John Wiley, 2003.
6. Herbert Schildt , *The Complete Reference in C++*, 5th Ed., TMH, 2012.

CO-PO Articulation Matrix Object Oriented Programming Using C++ Course (PC/CSE/2-T)

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1. **list** the concepts related to object-oriented paradigms. (LOTS: Level 1: Remember)	1	1	-	-	-	-	-	-	-	-	-	-	2	-	-
CO2. **distinguish** between structured and object-oriented approaches to programming. CO3. (LOTS: Level 2: Understand)	1	1	-	-	-	-	-	-	-	-	-	-	3	-	-
CO4. **Apply** object-oriented constructs for problem solving. (LOTS: Level 3: Apply)	2	1	-	-	2	-	-	-	-	-	-	-	3	-	-
CO5. **Detect** logical and run time errors and suggest appropriate modifications. CO6. (HOTS: Level 4: Analyse)	2	2	-	-	-	-	-	-	-	-	-	-	3	-	-
CO7. **Justify** the design of a program for a given problem. (HOTS: Level 5: Evaluate)	2	3	-	-	-	-	-	-	1	-	-	-	3	-	-
CO8. **Design** solutions to programming problems using multiple object-oriented programming constructs together. (HOTS: Level 6: Create)	3	3	1	-	2	-	-	-	1	-	-	-	3	-	-
Level of Attainments PC/CSE/2-T															

Discrete Mathematics

General Course Information

Course Code: PC/CSE/3-T Course Credits: 3 Type: Professional Core Contact Hours: 3 hours/week Mode: Lectures Examination Duration: 3 hours	**Course Assessment Methods (internal: 30; external: 70)** Two minor examinations (20 marks), Class Performance measured through percentage of lectures attended (4 marks), assignments (6 marks), and the end- semester examination (70 marks). For the end semester examination, nine questions are to be set by the examiner. A candidate is required to attempt 5 questions in all. All questions carry equal marks. Question number 1 will be compulsory and based on the entire syllabus. It will contain seven parts of 2 marks each. Question numbers 2 to 9 will be given by setting two questions from each of the four units of the syllabus. A candidate is required to attempt the remaining four questions by selecting one question from each of the four units.

Pre-requisites: Basic knowledge of Number Theory, Calculus and Algebra

About the Course:
Discrete Mathematics is a core and an essential course for every graduate in Computer Science and Engineering. This branch of mathematics mainly deals with discrete objects (as computer runs on discrete steps). It provides a mathematical language for computer science to resolve many real-world problems by incorporating different methods applicable to various discrete structures. This course introduces set theory, propositional calculus, algebraic structures, recurrence relations and graph theory.

Course Outcomes: By the end of the course a student would be able to:

CO1. **outline** various discrete structures and the related operations. (LOTS: Level 1: Remember)
CO2. **illustrate** different discrete structures with the help of examples. (LOTS: Level 2: Understand)
CO3. **apply** appropriate techniques to solve problems related to discrete structures. (LOTS: Level 3: Apply)
CO4. **justify** the solutions with the help of proofs. (HOTS: Level 5: Evaluate)
CO5. **combine** techniques related to discrete structures for solving real world problems. (HOTS: Level 6: Create)

Course Content

Unit I

Set Theory: Introduction to Set Theory, Venn Diagrams, Set Operations, Algebra of Sets, Duality, Finite, Infinite Sets and Counting Principle, Classes of Sets, Power Sets, Partitions, Multi Sets, Relations: Cartesian Product, Representation of Relations, Types of Relation, Equivalence Relations and Partitions, Partial Ordering Relations, Functions: Definition, Types of Functions, Composition of Functions, Inverse Function, Recursively Defined Functions.

Unit II

Logic and Propositional Calculus: Introduction, Propositions and Compound Propositions, Basic Logical Operations, Propositions and Truth Tables, Tautologies and Contradictions, Logical Equivalence, Algebra of Propositions, Conditional and Bi-conditional Statements, Algebraic Structures: Group Axioms, Monoid, Semi-Groups, Subgroups, Abelian Group, Cosets, Normal Subgroup, Cyclic Group, Permutation Group, Lagrange's Theorem, Homomorphism, Isomorphism, Automorphism, Rings, Integral Domains and Fields (Also, some basic and standard results related to Groups, Rings, ID and Fields).

Unit III

Recursion and Recurrence Relation: Polynomials and their evaluation, Sequences, Introduction to AP, GP and AG Series, Partial Fractions, Recurrence Relation, Linear Recurrence Relations with Constant Coefficients, Linear Homogeneous Recurrence Relations with Constant Coefficients, Particular Solution- Homogeneous Linear Difference Equations, Non-Homogeneous Linear Difference Equations, Total Solution, Generating Functions.

Unit IV

Graphs Theory: Introduction to Graphs, Multi Graph, Directed and Undirected Graphs, Subgraphs, Bipartite Graphs, Regular Graphs, Connected Graphs, Homomorphic and Isomorphic Graphs, Cut points and Bridges, Paths and Circuits, Euler Graph, Hamiltonian Graph, Planar Graph, Euler Formula, Weighted Graphs, Dijkstra's Shortest Path Algorithm for Weighted Graphs, Trees, Spanning Trees, Minimum Spanning Tree (Prim's and Kruskal's Algorithm).

T**ext and Reference Books:**

1. J.P. Trembley and R. Manohar, *Discrete Mathematical Structures with Applications to Computer Science*, Tata McGraw Hill – 13th reprint, 2012.
2. Kenneth H. Rosen, *Discrete Mathematics and its applications*, 6th Edition, Tata McGraw Hill, 2011.
3. Richard Johnsonbaugh, *Discrete Mathematics*, 6th Edition, Pearson Education Asia, 2011.
4. S. Lipschutz and M. Lipson, *Discrete Mathematics*, Tata McGraw Hill, 3rd Edition, 2010.
5. B. Kolman, R. C. Busby and S. C. Ross, *Discrete Mathematical structures*, 6th Edition, PHI, 2010.
6. C. L. Liu, *Elements of Discrete Mathematics*, Tata McGraw Hill, 3rd Edition, 2008.

CO-PO Articulation Matrix Discrete Mathematics Course (PC/CSE/3-T)

List of Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1. **Outline** various discrete structures and the related operations. (LOTS: Level 1: Remember)	1	--	--	--	--	--	--	--	--	--	--	--	1	1	1
CO2. **Illustrate** different discrete structures with the help of examples. (LOTS: Level 2: Understand)	1	--	--	--	--	--	--	--	--	--	--	--	1	2	1
CO3. **Apply** appropriate techniques to solve problems related to discrete structures. (LOTS: Level 3: Apply)	2	--	--	--	1	--	--	--	--	--	--	--	2	2	--
CO4. **Justify** the solutions with the help of proofs. (HOTS: Level 5: Evaluate)	3	1	--	--	2	--	--	--	--	--	--	--	3	--	--
CO5. **Combine** techniques related to discrete structures for solving real world problems. (HOTS: Level 6: Create)	3	2	--	--	2	--	--	--	1	--	--	1	3	--	--
Level of Attainments: PC/CSE/3-T															

Computer Organization and Architecture

General Course Information

Course Code: PC/CSE/4-T Course Credits: 3 Type: Professional Core Contact Hours: 3 hours/week Mode: Lectures Examination Duration: 3 hours	**Course Assessment Methods (internal: 30; external: 70)** Two minor examinations each of 20 marks, Class Performance measured through percentage of lectures attended (4 marks) Assignment and quiz (6 marks), and end semester examination of 70 marks. For the end semester examination, nine questions are to be set by the examiner. Question number one will be compulsory and based on the entire syllabus. It will contain seven short answers type questions each of marks 2. Rest of the eight questions are to be given by setting two questions from each of the four units of the syllabus. A candidate is required to attempt any other four questions selecting one from each of the remaining four units. All questions carry equal marks.

Pre-requisites: Digital Electronics and computer systems.

About the Course:
Computer Architecture and organization describes the role of instruction set architecture in digital computer, main memory, and input/output devices. It illustrates the simple data path and control design for processors. It helps to understand the different operations and concept of instructions. It would enable the students to learn the basic function and architecture of modern computer systems.

Course Outcomes: By the end of the course students will be able to:

CO1. **outline** the general concepts of digital electronics and computer organisation and architecture. (LOTS: Level 1: Remember)
CO2. **discuss** the basic components and their interfacing. (LOTS: Level 2: Understand)
CO3. **apply** instructions for performing different operations. (LOTS: Level 3: Apply)
CO4. **analyse** the effect of addressing modes on the execution time of a program. (HOTS: Level 4: Analyse)
CO5. **contrast** different types of memory, their architecture and access methods. (HOTS: Level 5: Evaluate)
CO6. **Design of** simple computer with different instruction sets. (HOTS: Level 6: Create)

Course Content

Unit I

Basic Principles: Boolean algebra and Logic gates, Combinational logic blocks (Adders, Subtractors, Multiplexers, Encoders, decoders, demultiplexers, K-Maps), Sequential logic blocks (Flip-Flops, Registers); Flynn's classification of computers (SISD, MISD, MIMD); CPU Architecture types: computer register, (accumulator, register, stack, memory/ register) detailed data path of a typical register-based CPU.

Unit II

Computer Organization: Store program control concept, Instruction codes, timing and control, instruction cycle; type of instructions: memory reference, register reference, I/O reference; Basics of Logic Design, accumulator logic, Control memory; Micro Programmed Control:

address sequencing, micro-instruction formats, micro-program sequencer, Implementation of control unit.

Unit III

Instruction Set Architecture & Parallelism: Instruction set based classification of processors (RISC, CISC, and their comparison); Stack Organization, Instruction Formats; addressing modes: register, immediate, direct, indirect, indexed; Operations in the instruction set: Arithmetic and Logical, Data Transfer, Control Flow; Types of interrupts; Introduction to Parallelism: Goals of parallelism (Exploitation of concurrency, throughput enhancement); Amdahl's law; Instruction level parallelism (pipelining, super scaling –basic features); Processor level parallelism (Multiprocessor systems overview).

Unit IV

Memory Hierarchy & I/O Techniques: The need for a memory hierarchy (Locality of reference principle, Memory hierarchy in practice: Cache, main memory and secondary memory, Memory parameters: access/ cycle time, cost per bit); Main memory (Semiconductor RAM & ROM organization, memory expansion, Static & dynamic memory types); Cache memory (Associative & direct mapped cache organizations; input-output interface, mode of transfer, DMA (Direct memory transfer).

T**ext and Reference Books:**

1. Mano, M. Morris, *Digital Logic and Computer Design*, Prentice Hall of India Pvt. Ltd., 1981.
2. M. Morris Mano, *Computer System Architecture*, Prentice Hall of India Pvt. Ltd., 1993.
3. Milles J. Murdocca, Vincent P. Heuring, *Computer Architecture and Organization, An Integrated Approach*, JohnWiley & Sons Inc., 2007.
4. William Stallings, 10th edition, *Computer Organization and Architecture*, Prentice Hall, 2016.
5. Heuring, V.P., Jordan, H.F., *Computer Systems Design and Architecture*, Addison Wesley,1997.
6. R.P Jain, *Modern Digital Electronics*, 3rd Edition, Tata McGraw Hill, 2003.

CO-PO Articulation Matrix Computer Organization and Architecture Course (PC/CSE/4-T)

List of Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1. **outline** the general concepts of digital electronics and computer organisation and architecture. (LOTS: Level 1: Remember)	1	-	-	-	-	-	-	-	-	-	-	-	2	-	-
CO2. **discuss** the basic components and their interfacing. (LOTS: Level 2: Understand)	1	-	-	-	-	-	-	-	-	-	-	-	3	-	-
CO3. **Apply** instructions for performing different operations. (LOTS: Level 3: Apply)	2	-	-	-	-	-	-	-	-	-	-	-	3	-	-
CO4. **Analyse** the effect of addressing modes on the execution time of a program. (HOTS: Level 4: Analyse)	2	2	-	1	-	-	-	-	-	-	-	1	3	-	-
CO5. **Contrast** different types of memory, their architecture and access methods. (HOTS: Level 5: Evaluate)	2	2	-	1	-	-	-	-	-	-	-	1	3	-	-
CO6. **Design** of simple computer with different instruction sets. (HOTS: Level 6: Create)	3	2	-	-	2	-	-	-	-	-	-	-	3	-	-
Level of Attainments PC/CSE/4-T															

Environmental Science

General Course Information

Course Code: MC/2-T Course Credits: 0 Type: Mandatory Course Contact Hours: 3 hours/week Mode: Lectures (L) Examination Duration: 3 hours	**Course Assessment Methods (internal: 30; external: 70)** Two minor examinations (20 marks), Class Performance measured through percentage of lectures attended (4 marks), assignments (6 marks), and the end-semester examination (70 marks). For the end semester examination, nine questions are to be set by the examiner. A candidate is required to attempt 5 questions in all. All questions carry equal marks. Question number 1 will be compulsory and based on the entire syllabus. It will contain seven parts of 2 marks each. Question numbers 2 to 9 will be given by setting two questions from each of the four units of the syllabus. A candidate is required to attempt the remaining four questions by selecting one question from each of the four units.

Pre-requisites: None

About the Course and its Outcomes:
This is a mandatory course to enhance the knowledge, skills and attitude of the graduating engineers to the environment. By studying this course students will understand our natural environment and its relationship with human activities.

Course outcomes: By the end of the course a student will be able to:

CO1. **state** the environment related issues and challenges in sustainable development
CO2. **demonstrate** the understanding of various environment hazards and means of protection against these hazards. (LOTS: Level 2: Understand)
CO3. **apply** irreplaceable tool to provide first-hand knowledge on various environmental aspects in the entire learning process. (LOTS: Level 3: Apply)
CO4. **analyze** impacts of human business and developmental activities on the environment. (HOTS: Level 4: analyze)
CO5. **design** and evaluate strategies for sustainable management of environmental eco-systems. (HOTS: Level 6: design)

Course content

Unit-I

Multidisciplinary nature of Environmental studies: Definition, scope and importance, need for public awareness; Concept, Structure and function of an ecosystem: Producers, consumers and decomposers, Energy flow in the ecosystem ,Ecological succession ,Food chains, Food webs and ecological pyramids; Introduction, types, characteristics features, structure and function of Forest ecosystem, Grassland ecosystem ,Desert ecosystem, Aquatic ecosystem (Ponds, Stream, lakes, rivers, oceans, estuaries); Biodiversity: Introduction, Definition: genetic, species and ecosystem diversity, Bio-geographical classification of India, Value of biodiversity: consumptive use, productive use, social ethical, aesthetic and option values; Biodiversity at global, national and local level, India as a mega-diversity nation, Hot-spot of biodiversity, Threats to biodiversity: habitat loss, poaching of wildlife, man-wildlife conflicts, Endangered and endemic species of India, Conservation of biodiversity: In-situ and Ex-situ conservation of biodiversity.

Unit-II

Renewable and non-renewable resources, Natural resources and associated problems ,Forest resources: Use and over-exploitation, deforestation, case studies, Timber extraction, mining, dams and their effects on forests and tribal people; Water resources: Use and over utilization of surface and ground water, floods, droughts conflicts over water, dams benefits and problems; Mineral resources: Use and exploitation, environmental effects of extracting and mineral resources; Food resources: World food problem, changes caused by agriculture and overgrazing, effects of modern agriculture, fertilizer-pesticide problems, water logging, salinity; Energy resources: Growing energy needs, renewable and non-renewable energy sources, use of alternate energy sources, case studies; Land resources: Land as a resource, land degradation, main induced landslides, soil erosion and desertification, Role of an individual in conservation of natural resources, Equitable use of resources for suitable lifestyle.

Unit-III

Definition of Environment Pollution; Causes, effects and control measures of: Air Pollution, Water Pollution, Soil pollution, Marine pollution, Noise pollution, Thermal pollution, Nuclear hazards; Solid waste Management: Causes effects and control measures of urban and industrial wastes; Role of and individual in prevention of pollution, Pollution case studies; Disaster management: floods, earthquake, cyclone and landslides; Climate change, global warming, acid rain, ozone layer depletion, nuclear accidents and holocaust, Case studies; different laws related to environment: Environment Protection Act, Air (Prevention and Control of Pollution) Act, Water (Prevention and Control of Pollution) Act, Wildlife Protection Act, Forest Conservation Act.; Issues involved in enforcement of environmental legislation, Public awareness

Unit-IV

Social issues and the Environment: From unsustainable to Sustainable development, Urban problems related to energy; Water conservation, rain water harvesting, watershed management; Resettlement and rehabilitation of people; its problem and concern, case studies; Environment ethics: Issues and possible solutions; Wasteland reclamation; Consumerism and waste products; Human Population growth, variation among nation, Population explosion- Family Welfare Programme, Environment and human health , Human Rights, Value Education, HIV/AIDS, Women and Child Welfare, Role of Information Technology in Environment and human health, Case Studies.

Field Work: Visit to a local area to document environmental assets- river/forest/ grassland/hill/mountain; Study of simple ecosystems – ponds, river, hill slopes etc; Study of common plants, insects, birds; Visit to a local polluted site- Urban/Rural/Industrial/Agricultural.

T**ext and Reference Books:**

1. Erach Bharucha, *Environmental Studies for Undergraduate Courses*, University press pvt. Ltd. (India), 2005.
2. Dr. D. D. Mishra, *Fundamental concepts in Environmental studies*, S. Chand publications, 2008.
3. Dr. S. V.S. Rana, *Essentials of Ecology and Environmental Science*, PHI Learning Pvt. Ltd Delhi, 2013.
4. Anil Kumar De, *Environmental Chemistry*, Wiley Eastern Limited, 1994.

5. T. G. Miller, *Environmental Science,* Wadsworth Publishing Co, 13th4edition, 2013.
6. P. D. Sharma, *Ecology and Environment*, Rastogi publications, 13th edition, 2003.

CO-PO Articulation Matrix Environmental Studies (MC/2-T)

List of Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1. state the environment issues and challenges for sustainable development	-	1	-	-	-	-	-	-	-	-	-	-			
CO2. demonstrate the understanding of various environment hazards and means of protection against these hazards. (LOTS: Level 2: Understand)	-	-	-	2	-	-	-	-	-	-	-	-			
CO3. apply irreplaceable tool to provide a first-hand knowledge on various environmental aspects in the entire learning process. (LOTS: Level 3: Apply)	-	-	-	-	3	-	-	-	-	-	-	2			
CO4. analyze impacts of human business and developmental activities on the environment. (HOTS: Level 4: analyze)	1	2	-	-	-	2	-	3	-	3	-	-			
CO5. design and evaluate strategic methods for sustainable management of environmental eco-systems. (HOTS: Level 6: design)	1	2	2	-	-	-	3	-	3	-	2	-			
Level of Attainments MC/2-T															

Data Structures and Algorithms using C/C++Lab.

General Course Information

Course Code: PC/CSE/1-P Course Credits: 2 Type: Professional Core Lab. Course Contact Hours: 4 hours/week Mode: Lab practice and assignments	**Course Assessment Methods (internal: 50; external: 50)** The internal and external assessment is based on the level of participation in lab. Sessions and the timely submission of lab experiments/assignments, the quality of solutions designed for the assignments, the performance in VIVA-VOCE, the quality of lab. file and ethical practices followed. The internal examination is conducted by the course coordinator. The external examination is conducted by external examiner appointed by the Controller of Examination in association with the internal examiner appointed by the Chairperson of the Department.

Pre-requisites: Programming in C language.

About the Course:
This lab. course involves implementation of basic and advance data structures and various operations on these data structures. The objective of the lab course is to train the students to solve the problems related to data structures and choose the appropriate data structure for solving computational problem efficiently.

Course Outcomes: By the end of the lab course a student would be able to:

CO1. **Implement** various data structures and the related operations. (LOTS: Levels 3: Apply)
CO2. **Analyse** space and time complexity of algorithms. (HOTS: Level 4: Analyse)
CO3. **Compare** solutions on the basis of the appropriateness of data structure used and the efficiency of the operations implemented. (HOTS: Level 5: Evaluate)
CO4. **Integrate** knowledge of data structures to solve real world problems related to data structure and algorithms. (HOTS: Level 6: Create)
CO5. **Create** written records for the given assignments with problem definition, design of solution and conclusions. (HOTS: Level 6: Create)
CO6. **Demonstrate** ethical practices while solving problems individually or in groups (LOTS: Level 3: Apply).

List of experiments/assignments

1. Two assignments related to creating and manipulating matrices and linear lists.
2. Two assignments associated with linked list, operations on linked lists and their applications.
3. Two assignments on array and linked implementation of stacks and queues.
4. Two assignments on trees and their applications.
5. Two assignments on graphs and their applications.
6. Two assignments on different searching and sorting methods along with their complexity analysis.
7. One assignment on challenging problems on data structures to be given in groups.

Note:

The actual experiments/assignments will be designed by the course coordinator. One assignment should be designed to be done in groups of two or three students. The assignments must meet the objective of the course and the levels of the given course outcomes. The list of assignments and schedule of submission will be prepared by the course coordinator at the beginning of the semester.

CO-PO Articulation Matrix Data Structures and Algorithms Lab. Course (PC/CSE/1-P)

List of Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1. **Implement** various data structures and the related operations. (LOTS: Levels 3: Apply)	2	-	-	-	1	-	-	-	2	-	-	-	3	-	-
CO2. **Analyse** space and time complexity of algorithms. (HOTS: Level 4: Analyse)	2	2	-	-	1	-	-	-	1	-	-	-	3	-	-
CO3. **compare** solutions on the basis of the appropriateness of data structure used and the efficiency of the operations implemented. (HOTS: Level 5: Evaluate)	2	2	-	-	1	-	-	-	1	-	-	-	3	-	-
CO4. **integrate** knowledge of data structures to solve real world problems related to data structure and algorithms. (HOTS: Level 6: Create)	3	2	3	-	-	-	-	-	3	-	-	-	3	-	-
CO5. **Create** written records for the given assignments with problem definition, design of solution and conclusions. (HOTS: Level 6: Create)	-		-	-	-	-	-	-	-	3	-	-	-	-	-
CO6. **Demonstrate** ethical practices while solving problems individually or in groups (LOTS: Level 3: Apply).	-		-	-	-	-	-	3	-	-	-	3	-	-	-
Level of Attainments: PC/CSE/1-P															

Object Oriented Programming using C++ Lab.

General Course Information

Course Code: PC/CSE/2-P Course Credits: 2 Type: Professional Core Lab. Course Contact Hours: 4 hours/week Mode: Lab practice and assignments	**Course Assessment Methods (internal: 30; external: 70)** The internal and external assessment is based on the level of participation in lab. sessions and the timely submission of lab experiments/assignments, the quality of solutions designed for the assignments, the performance in VIVA-VOCE, the quality of lab. file and ethical practices followed. The internal examination is conducted by the course coordinator. The external examination is conducted by external examiner appointed by the Controller of Examination in association with the internal examiner appointed by the Chairperson of the Department.

Pre-requisites: Problem solving using C Lab.

About the course:
The lab course provides the opportunity to students to solve problems using Object Oriented Framework in C++ language. This includes implementing the concepts of data abstraction, data hiding, and encapsulation, reuse of code and, compile and runtime polymorphism.

Course Outcomes: By the end of the course students will be able to:

CO1. **implement** problems with object-oriented framework. (LOTS: Level 3: Apply)
CO2. **analyse** the structure of programs for modular design. (HOTS: Level 4: Analyse)
CO3. **evaluate** robustness of a program by testing it on test/use cases. (HOTS: Level 5: Evaluate)
CO4. **design** class hierarchies for implementing inheritance/polymorphism. (HOTS: Level 6: Create)
CO5. **create** a lab record of assignments including problem definitions, design of solutions and conclusions. (HOTS: Level 6: Create)
CO6. **demonstrate** ethical practices and solve problems individually or in a group. (LOTS: Level 3: Apply)

List of assignments

1. Create two classes **DM** and **DB** which store the value of distances. **DM** stores distances in meters and centimeters and **DB** in feet and inches. Write a program that can read values for the class objects and add one object of **DM** with another object of **DB**. Use a friend function to carry out the addition operation. The object that stores the results maybe a **DM** object or **DB** objects, depending on the units in which the result is required. The display should be in the format of feet and inches or meters and centimeters depending on the object on display.
2. Create a class rational which represents a numerical value by two double values-NUMERATOR & DENOMINATOR. Include the following public member Functions:
    - constructor with no arguments (default).
    - constructor with two arguments.
    - void reduce () that reduces the rational number by eliminating the highest common factor between
    - the numerator and denominator.
    - Overload + operator to add two rational number.

- Overload >> operator to enable input through cin.
- Overload << operator to enable output through cout.

Write the main () function to test all the functions in the class.

3. A hospital wants to create a database regarding its indoor patients. The information to be stored includes
   a) Name of the patient b) Date of admission c) Disease d) Date of discharge
   Create a structure to store the date (year, month and day as its members). Create a base class to store the above information. The member function should include functions to enter information and display a list of all the patients in the database. Create a derived class to store the age of the patients. List the information about all the pediatric patients (less than twelve years in age).
4. Make a class **Employee** with a name and salary. Make a class **Manager** inherited from **Employee**. Add an instance variable named department of type string. Supply a method to **to String** that prints the manager's name, department and salary. Make a class **Executive** inherited from **Manager**. Supply a method **to String** that prints the string **"Executive"** followed by the information stored in the **Manager** superclass object. Supply a test program that tests these classes and methods.
5. Imagine a tollbooth with a class called 'toll Booth'. The two data items are of type unsigned int to hold the total number of cars, and a type double to hold the total amount of money collected. A constructor initializes both to 0. A member function called 'payingCar()' increments the car total and adds 0.50 to the cash total. Another function, called 'nopayCar ()', increments the car total but adds nothing to the cash total. Finally, a member function called displays the two totals. Include a program to test this class. This program should allow the user to push one key to count a paying car, and another to count a nonpaying car. Pushing the ESC kay should cause the program to print out the total cars and total cash and then exit.
6. Write a function called 'revers_it()' that reverses a string (an array of char). Use a for loop that swaps the first and last characters, then the second and next to last characters and so on. The string should be passed to 'revers_it ()' as an argument. Write a program to exercise 'revers_it ()'. The program should get a string from the user call of 'revers_it () function and print out the result. Use an input method that allows embedded blanks.
7. Write a program related to file handling with all the exception handling provisions.
8. C++ program to write and read time in/from binary file using fstream. Use exception handling wherever possible.
9. Write a program to implement string class using STL.
10. Write a program to implement run time polymorphism.

Note:

The experiments/assignments may vary from session to session and will be designed by the course coordinator. The assignments must meet the objective of the course and the levels of the given course outcomes. The course coordinator will provide the schedule for submission of the assignment.

CO-PO Articulation Matrix Object Oriented Programming using C++ Lab. (PC/CSE/2-P)

List of Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1. **Implement** problems with object-oriented framework. (LOTS: Level 3: Apply)	2	2	-	-	1	-	-		2	-	-	2	3	-	-
CO2. **Analyse** the structure of programs for modular design. (HOTS: Level 4: Analyse)	2	2	-	-	2	-	-	-	-	-	-	-	3	-	-
CO3. **evaluate** robustness of a program by testing it on test/use cases. (HOTS: Level 5: Evaluate)	2	2	-	-	2	-	-	-	-	-	-	-	3	-	-
CO4. **Design** class hierarchies for implementing inheritance/polymorphism. (HOTS: Level 6: Create)	3	-	1	-	2	-	-	-	-	-	-	2	3	-	-
CO5. **Create** a lab record of assignments including problem definitions, design of solutions and conclusions. (HOTS: Level 6: Create)	-	-	-	-	-	-	-		-	3	-	-	-	-	-
CO6. **Demonstrate** ethical practices and solve problems individually or in a group. (LOTS: Level 3: Apply)	-	-	-	-	-	-	-	3	3	-	-	3	-	-	-
Level of Attainments: PC/CSE/2-P															

Detailed Syllabus of

B.Tech.(CSE)

4th Semester

General Course Information:

Course Code: PC/CSE/5-T Course Credits: 3 Type: Professional Core Contact Hours: 3 Mode: Lecture (L) Examination Duration: 3 Hours	**Course Assessment Methods (internal: 30; external: 70)** Two minor examinations (20 marks), Class Performance measured through percentage of lectures attended (4 marks), assignments (6 marks), and the end-semester examination (70 marks). For the end semester examination, nine questions are to be set by the examiner. A candidate is required to attempt 5 questions in all. All questions carry equal marks. Question number 1 will be compulsory and based on the entire syllabus. It will contain seven parts of 2 marks each. Question numbers 2 to 9 will be given by setting two questions from each of the four units of the syllabus. A candidate is required to attempt the remaining four questions by selecting one question from each of the four units.

Pre-requisites: The students are expected to have a strong background in the Computer Organization Digital System Design.

About the Course

By the end of the course students will be able to:

CO1. **Outline** the architecture of 8085 and 8086 Microprocessor. (LOTS: Level 1: Remember)

CO2. **Discuss** the basic principles of addressing modes, pin diagrams. (LOTS: Level 2: Understand)

CO3. **Describe** the functionality of various peripheral chip (LOTS: Level 2: Understand)

CO4. **Apply** the concepts of interfacing of Memory, Input/output with Microprocessor. (LOTS: Level 3: Apply)

CO5. **Compare** and **contrast** the working of 8085 and 8086 microprocessors. (HOTS: Level 5: Evaluate)

CO6. **Develop** Assembly Language programs for 8085 and 8086 microprocessors. (HOTS: Level 6: Create)

Course Contents

Unit-I

Introduction to microprocessor, 8085 microprocessor architecture, instruction set, interrupt structure, Architecture of 8086, block diagram of 8086, details of sub-blocks such as EU, BIU; memory segmentation and physical address computations, program relocation.

Unit II

Addressing modes, instruction formats, pin diagram and description of various signals, Instruction execution timing, assembler instruction format, data transfer instructions, arithmetic instructions, branch instructions, looping instructions, NOP and HLT instructions, flag manipulation instructions, logical instructions, shift and rotate instructions, directives and operators.

Unit III

Assembler directives, Programming with an assembler, Programming examples, coding style, the art of assembly language programming.
Software Development with Interrupts, Introduction to Stack, Stack Structure of 8086, Introduction to Subroutines, Recursion, MACROS. BIOS (Basic Input/Output System), DOS (Disk Operating System.

Unit IV

The 8255 PPI chip: Architecture, control words, modes and examples.
Introduction to DMA process, 8237 DMA controller.
8259 Programmable interrupt controller, Programmable interval timer chips.

T**ext and Reference Books:**

1. Ramesh S Gaonkar; *Microprocessor Architecture, Programming & Applications with 8085*, Wiley Eastern Ltd., 5th edition, 2002.
2. Brey, *The Intel Microprocessors 8086- Pentium processor*, PHI, 8th edition, 2009.
3. Douglas V Hall; *Microprocessors and Interfacing*, TMH, 2000.
4. Triebel & Singh; *The 8088 & 8086 Microprocessors-Programming, interfacing, Hardware & Applications*, PHI, 4th edition, 2003.
5. Yu-Chang Liu & Glenn A Gibson; *Microcomputer systems: the 8086/8088 Family: architecture, Programming & Design,* PHI, 1986.

CO-PO Articulation Microprocessor and Interfacing Course (PC/CSE/5-T)

List of Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1. **Outline** the architecture of 8085 and 8086 Microprocessor. (LOTS: Level 1: Remember)	1	--	--	--	--	--	--	--	--	--	--	--	3	--	--
CO2. **Discuss** the basic principles of addressing modes, pin diagrams. (LOTS: Level 2: Understand)	2	--	--	--	--	--	--	--	--	--	--	--	3	--	--
CO3. **describe** the functionality of various peripheral chip (LOTS: Level 2: Understand)	2	1	--	--	--	--	--	--	--	--	--	--	3	--	--
CO4. **apply** the concepts of interfacing of Memory, Input/output with Microprocessor. (LOTS: Level 3: Apply)	2	1	--	--	--	--	--	--	--	--	--	--	3	--	--
CO5. **compare** and **contrast** the working of 8085 and 8086 microprocessors. (HOTS: Level 5: Evaluate)	2	--	1	1	2	--	--	--	--	--	--	1	3	--	--
CO6. **develop** Assembly Language programs for 8085 and 8086 microprocessors. (HOTS: Level 6: Create)	3	2	2	--	2	--	--	--	1	--	--	1	3	--	--
Level of Attainments PC/CSE/5-T															

Computer Networks

General Course Information

Course Code: PC/CSE/6-T Course Credits: 3 Type: Professional Core Contact Hours: 3 hours/week Mode: Lectures (L) Examination Duration: 3 hours	**Course Assessment Methods (internal: 30; external: 70)** Two minor examinations (20 marks), Class Performance measured through percentage of lectures attended (4 marks), assignments (6 marks), and the end-semester examination (70 marks). For the end semester examination, nine questions are to be set by the examiner. A candidate is required to attempt 5 questions in all. All questions carry equal marks. Question number 1 will be compulsory and based on the entire syllabus. It will contain seven parts of 2 marks each. Question numbers 2 to 9 will be given by setting two questions from each of the four units of the syllabus. A candidate is required to attempt the remaining four questions by selecting one question from each of the four units.

Pre-requisites: Basic knowledge of Digital and Analog Communication.

About the Course:

This course has been designed with an aim to provide students with an overview of the concepts and fundamentals of data communication and computer networks. The learner is given an opportunity to grasp various algorithms for routing of data, forwarding data and switching the data from hop to hop. Layered Architecture adds value to the subject contents.

Course Outcomes: By the end of the course students will be able to:

CO1. **outline** various models, topologies and devices of Computer Networks. (LOTS: Level 1: Remember)

CO2. **explain** the functions of various layers in Network Reference Model. (LOTS: Level 2: Understand)

CO3. **apply** different network concepts in various network communication protocols. (LOTS: Level 3: Apply)

CO4. **analyse** performance of various protocols in different scenarios. (HOTS: Level 4: Analyse)

CO5. **design** network for an organisation. (HOTS: Level 6: Create)

Course content

Unit I

Data communication: Components, Data representation and Data flow; Network: Uses, Topologies, Network Services, OSI and TCP/IP Reference Models; Network categories: LAN, MAN, WAN; Guided Transmission Media, Wireless Transmission Media, Switching Techniques: Circuit Switching, Packet Switching, Message Switching, Networking Devices**:** Hubs, Repeaters, Bridges, Modems, Switches, Routers, and Gateways.

Unit II

Data Link Layer-design issues, Framing & Error Handling: Framing Protocols, Error detection and correction mechanisms; Flow Control Protocols: Stop-and-wait, Sliding Window protocols: Go-back-N and Selective Repeat; Medium Access sub layer: Channel allocation methods,

Multiple Access Communication: Random Access-ALOHA, Slotted-ALOHA, CSMA, CSMA-CD, LAN Standards: Ethernet, Fast Ethernet & Gigabit Ethernet.

Unit III

Network Layer-Design issues, store and forward packet switching connection less and connection-oriented networks, Routing algorithms: optimality principle, shortest path, flooding, Distance Vector Routing, Count to Infinity Problem, Link State Routing, Hierarchical Routing, Congestion control algorithms, admission control.
Internetworking: IPV4 and IPV6, IP Addressing (Classful Addressing, Private IP Addresses, Classless Addressing, Sub-netting), ARP, RARP, ICMP, Internet Routing Protocol.

Unit IV

Transport Layer: Transport layer Services: Addressing, Multiplexing, Flow control, Buffering and Error control. Internet Transport Protocols: UDP, TCP, TCP Segment, TCP Connection.
Application Layer: Introduction to DNS, FTP, TELNET, HTTP, SMTP, Electronic Mail, WWW and Multimedia.

Text and Reference Books:

1. Andrew S Tanenbaum, *Computer Networks*, 5th Edition, Pearson publications, 2010.
2. Forouzan, *Data Communication and networking* ,5th Edition, Tata McGrawHill, 2012.
3. William Stalling, *Data & Computer Communication* 6th edition, LPE Pearson Education, 2013.
4. Todd Lammle, *CCNA Study Guide*, 6th Edition, 2013.
5. RFCs and Internet Drafts available from Internet Engineering Task Force.

CO-PO Articulation Matrix Computer Networks Course (PC/CSE/6-T)

List of Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1. **outline** various models, topologies and devices of Computer Networks. (LOTS: Level 1: Remember)	1	-	-	-	-	-	-	-	-	-	-	-	-	3	-
CO2. **explain** the functions of various layers in Network Reference Model. (LOTS: Level 2: Understand)	1	-	-	-	-	-	-	-	-	-	-	-	-	3	-
CO3. **Apply** different network concepts in various network communication protocols. (LOTS: Level 3: Apply)	2	1	-	-	2	-	-	-	-	-	-	-	-	3	-
CO4. **Analyse** performance of various protocols in different scenarios. (HOTS: Level 4: Analyse)	2	2	2	1	2	-	-	-	-	-	-	-	-	3	-
CO5. **Design** network for an organisation. (HOTS: Level 6: Create)	3	2	2	-	2	-	-	-	-	-	-	-	-	3	-
Level of Attainments PC/CSE/6-T															

Database Management System

General Course Information

Course Code: PC/CSE/7-T Course Credits: 3 Type: Professional Core Contact Hours: 3 hours/week Mode: Lectures (L) Exam Duration: 3 hours	**Course Assessment Methods (internal: 30; external: 70)** Two minor examinations (20 marks), Class Performance measured through percentage of lectures attended (4 marks), assignments (6 marks), and the end-semester examination (70 marks). For the end semester examination, nine questions are to be set by the examiner. A candidate is required to attempt 5 questions in all. All questions carry equal marks. Question number 1 will be compulsory and based on the entire syllabus. It will contain seven parts of 2 marks each. Question numbers 2 to 9 will be given by setting two questions from each of the four units of the syllabus. A candidate is required to attempt the remaining four questions by selecting one question from each of the four units.

Prerequisite: Knowledge of UNIX, Windows, a programming language and data structures

About the Course:
This course includes a detailed coverage of principles of database design and models. Students learn querying a database using SQL, normalization techniques, transaction processing etc.

Course Outcomes: By the end of the course students will be able to:

CO1. **describe** fundamental elements of Database Management System. (LOTS: Level 1: Remember)
CO2. **discuss** principles of relational Database modelling. (LOTS: Level 2: Understanding)
CO3. **apply** SQL for designing queries for Relational Databases. (LOTS: Level 3: Apply)
CO4. **contrast** various concurrency control and recovery techniques with concurrent transactions in DBMS. (HOTS: Level 5: Evaluate)
CO5. **design** models of databases using ER modelling and normalization for real life applications. (HOTS: Level 6: Create)

Course Content

Unit – 1

Overview: Overview of File Systems and Database Systems, Characteristics of the Data Base Approach, Database users, Advantages and Disadvantages of a DBMS, Responsibility of Database Administrator.
Data Base Systems Concepts and Architecture: DBMS architecture and various views of Data, Data Independence, Database languages, Data Models: Relational Database Model, Hierarchical Data Model, Network Data Model, Schemas and Instances.

Unit – 2

E-R Model: Entity Types, Attributes & Keys, Relationships, Roles and Structural Constraints, E-R Diagrams, Reduction of an E-R Diagram to Tables. Relational Model and Query Language: Overview of Relational Database, Key Integrity Constraints, Relational Algebra, Relational Calculus, SQL fundamentals, Basic Operators, Missing information and NULL values, Advanced SQL features

Unit - 3

Relational Database Design: Overview of normalization, Database Anomalies, Candidate and Super Key, Functional Dependencies, Integrity Constraints, Decomposition, Normal forms: First, Second, Third Normal, Boyce Codd, Normal Form, Multi-valued Functional Dependencies and Fourth Normal Form, Join Dependencies and Fifth Normal Form, Denormalization.

Unit - 4

Concurrency Control Techniques: Overview of database Transactions, Transaction states, ACID properties of a Transaction, Transaction Recovery, Concurrency Control, Locking Techniques, Time-stamp ordering, Multi-version Techniques, Deadlock, Recovery Techniques in centralized DBMS.
DDBMS Design: Replication and Fragmentation Techniques.

Text and Reference Books:

1. Elmasri, R., and Navathe, S. B., *Fundamentals of Database Systems*, 3rd Edition, Addison Wesley, 2002.
2. Silberschatz, A., Korth, H. F., and Sudarshan, S., Database *System Concepts,* McGraw Hill, 2011.
3. Pannerselvam R., *Database Management Systems*, 2nd Edition, PHI Learning, 2011.
4. Desai, B. C., *An Introduction to Database System*, Galgotia Publication, 2010.
5. Leon, A., and Leon, M., *Database Management Systems*, Ist Edition, Vikas Publishing, 2009.
6. Mata-Toledo, R., Cushman, P., Sahoo, D., *Database Management Systems*, Schaums' Outline series, TMH, 2007.

CO-PO Articulation Matrix Database Management System Course (PC/CSE/7-T)

List of Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1. **Describe** fundamental elements of Database Management System. (LOTS: Level 1: Remember)	1	--	--	--	--	--	--	--	--	--	--	--	3	--	--
CO2. **discuss** principles of relational Database modeling. (LOTS: Level 2: Understanding)	1	--	--	--	--	--	--	--	--	--	--	--	3	--	--
CO3. **Apply** SQL for designing queries for Relational Databases. (LOTS: Level 3: Apply)	1	--	--	_	2	--	--	--	--	--	--	--	3	--	--
CO4. **contrast** various concurrency control and recovery techniques with concurrent transactions in DBMS. (HOTS: Level 5: Evaluate)	1	2	--	--	--	--	--	--	--	--	--	--	3	--	--
CO5. **Design** models of databases using ER modelling and normalization for real life applications. (HOTS: Level 6: Create)	3	2	3	1	2	--	--	--	--	--	--	--	3	--	--
Level of Attainments PC/CSE/7-T															

Analysis and Design of Algorithms

General Course Information

Course Code: PC/CSE/8-T Course Credits: 3 Type: Professional Core Contact Hours: 3 hours/week Mode: Lectures (L) Examination Duration: 3 hours	**Course Assessment Methods (internal: 30; external: 70)** Two minor examinations (20 marks), Class Performance measured through percentage of lectures attended (4 marks), assignments (6 marks), and the end-semester examination (70 marks). For the end semester examination, nine questions are to be set by the examiner. A candidate is required to attempt 5 questions in all. All questions carry equal marks. Question number 1 will be compulsory and based on the entire syllabus. It will contain seven parts of 2 marks each. Question numbers 2 to 9 will be given by setting two questions from each of the four units of the syllabus. A candidate is required to attempt the remaining four questions by selecting one question from each of the four units.

Pre-requisites: Knowledge of Data Structure and a Programming Language

About the Course:
This Course focus on effective and efficient design of algorithms. In this course various algorithm design techniques and their analysis is to be studied. After studying this course, a student is expected to apply better techniques for solving computational problems efficiently and prove it analytically.

Course Outcomes: By the end of the course students will be able to:

- CO1. **state** terminology and concepts algorithmic techniques. (LOTS: Level 1: Remember)
- CO2. **discuss** various algorithmic techniques. (LOTS: Level 2: Understand)
- CO3. **apply** appropriate algorithmic techniques to solve computational problems. (LOTS: Level 3: Apply)
- CO4. **analysing** algorithms for their efficiency by determining their complexity. (HOTS: Level 4: Analyse)
- CO5. **compare** the pros and cons of applying the different algorithmic techniques to solve problems. (HOTS: Level 5: Evaluate)
- CO6. **formulate** efficient and effective algorithmic solutions for different real-world problems. (HOTS: Level: 6 Create)

Course Content

Unit I

Algorithms, Algorithms as a technology, Insertion sort, analyzing algorithms, asymptotic notations, Divide and Conquer: General method, binary search, merge sort, quick sort, Strassen‟s matrix multiplication algorithms and analysis of algorithms for these problems.

Unit II

Sorting and Data Structures: Heapsort, Hash Tables, Red and Black Trees, Greedy Method: General method, knapsack problem, minimum spanning trees, single source paths and analysis of these problems.

Unit III

Dynamic Programming: General method, matrix chain multiplication, longest common subsequence, optimal binary search trees,
Back Tracking: General method, 8 queen's problem, graph colouring, Hamiltonian cycles, Analysis of these problems.

Unit IV

Branch and Bound: Method, O/I knapsack and traveling salesperson problem, NP Completeness: Polynomial time, NP-completeness and reducibility, NP-complete problems.

Text and Reference Books:

1. Thomas H. Cormen, Charles E. Leiserson, Ronalde L. Rivest, Clifford Stein, *Introduction to Algorithms*, MIT press, 3rd Edition, 2009.
2. Ellis Horowitz, Satraj Sahni, Sanguthevar Rajasekaran, *Fundamental of Computer Algorithms*, Galgotia publication Pvt. Ltd., 1999.
3. S. Dasgupta, C. Papadimitriou, and U. Vazirani, *Algorithms*, McGraw-Hill Higher Education, 2006.

CO-PO Articulation Matrix Analysis and Design of AlgorithmsCourse (PC/CSE/8-T)

List of Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1. **state** terminology and concepts algorithmic techniques. (LOTS: Level 1: Remember)	1	--	--	--	--	--	--	--	--	--	--	--	2	--	--
CO2. **discuss** various algorithmic techniques. (LOTS: Level 2: Understand)	1	--	--	--	--	--	--	--	--	--	--	--	3	--	--
CO3. **Apply** appropriate algorithmic techniques to solve computational problems. (LOTS: Level 3: Apply)	2	--	1	--	--	--	--	--	--	--	--	--	3	2	2
CO4. **analysing** algorithms for their efficiency by determining their complexity. (HOTS: Level 4: Analyse)	3	2	1	--	2	--	--	--	--	--	--	--	3	2	2
CO5. **compare** the pros and cons of applying the different algorithmic techniques to solve problems. (HOTS: Level 5: Evaluate)	3	2	1	--	--	--	--	--	--	--	--	--	3	2	2
CO6. **formulate** efficient and effective algorithmic solutions for different real-world problems. (HOTS: Level: 6 Create)	3	3	2	2	--	--	--	--	--	--	--	--	3	2	2
Level of Attainments PC/CSE/8-T															

Software Engineering

General Course Information

Course Code: PC/CSE/9-T Course Credits: 3 Type: Professional Core Contact Hours: 3hours/week Mode: Lectures (L) Examination Duration: 3 hours	**Course Assessment Methods (internal: 30; external: 70)** Two minor examinations (20 marks), Class Performance measured through percentage of lectures attended (4 marks), assignments (6 marks), and the end-semester examination (70 marks). For the end semester examination, nine questions are to be set by the examiner. A candidate is required to attempt 5 questions in all. All questions carry equal marks. Question number 1 will be compulsory and based on the entire syllabus. It will contain seven parts of 2 marks each. Question numbers 2 to 9 will be given by setting two questions from each of the four units of the syllabus. A candidate is required to attempt the remaining four questions by selecting one question from each of the four units.

Pre-requisites: Knowledge of algorithms, flow charts and a programming language.

About the Course:

Software Development is generally a quite complex and time-consuming process. Moreover, depending on the nature and complexity of the software requirements, Software Engineering plays an important role. This course will help the students to understand the systematic approach to requirement analysis, design, development, operations and maintenance of software systems. Besides this, it also guides students in developing the optimal software systems.

Course Outcomes: By the end of the course students will be able to:

CO1. **define** the various concepts related to software engineering. (LOTS: Level 1: Remember)
CO2. **demonstrate** the use of stages of various Software Life Cycle Models. (LOTS: Level 2: Understanding)
CO3. **apply** the Software Requirement Analysis and Software Design Process. (LOTS: Level 3: Apply)
CO4. **analyse** the size, cost, complexity, reliability, quality and maintenance of a software system. (HOTS: Level 4: Analyse)
CO5. **construct** software model according to the requirements of a customer. (HOTS: Level 6: Create)

Course Content

Unit I

Introduction: Software Crisis, Software Process, Evolution of Software Engineering, Software Characteristics, Software Metrics and SDLC. Software Life Cycle Models: Water Fall Model, Increment Process Model, Evolutionary Process Models, Unified Process. Selection of Life Cycle Model.

Software Requirements, Analysis and Specifications: Requirement Engineering, Requirements Elicitation, Requirements Analysis: Data Flow Diagram, Data Dictionary, Entity-Relationship Diagrams, Decision Table, Decision Tree and Structured Charts. Requirements Documentation and Requirements validation.

Unit II

Software Project Management: Size Estimation, Cost Estimation, Constructive Cost Model (COCOMO), Putnam Resource Allocation Model. Software Risk Management: Software Risks, Risk Identification, Risk Projection, Risk Refinement, Risk Mitigation, Monitoring, and Management, RMMM Plan.

Unit III

Software Design: Software Design Fundamentals, Modularity, Design Principles, Strategy of Design, Function Oriented Design, and Object-Oriented Design, IEEE Recommended Practice for Software Design Descriptions.
Software Quality: Basic Concepts, ISO 9126, McCall's Quality Factors, Software Quality Assurance, SQA Activities, Software Review Process, Formal Technical Review, ISO 9000 Quality Standards, and CMM.

Unit IV

Software Testing: Testing fundamentals, Verification and Validation, Test Plan, Test Case, Levels of Software Testing: Unit Testing, Integration Testing, Top Down and Bottom-up Testing Integration Testing, Alpha and Beta Testing, System Testing, White Box Testing and Black Box Testing, Debugging and Software Testing Tools.
Maintenance and Reengineering**:** Software Maintenance, Software Supportability, Reengineering, Business Process Reengineering, Software Reengineering, Reverse Engineering, Restructuring, Forward Engineering and The Economics of Reengineering.

T**ext and Reference Books:**

1. K. K. Aggarwal and Yogesh Singh, *Software Engineering*, 3rd Edition, New Age International Publishers Ltd., Reprint 2014.
2. Roger S. Pressman, *Software Engineering: A Practitioners Approach* 7th Edition, Mc Graw Hill Education, 2014.
3. Rajib Mall, *Fundamental of Software Engineering*, Prentice Hall India, 2004.
4. Pankaj Jalote, *An integrated Approach to Software Engineering*, 3rd Edition, Narosa Publications, 2014.
5. Ian Sommerville, *Software Engineering*, 10th Edition, Addison-Wesley, 2015.
6. Carlo Ghezzi, Mehdi Jazayeri and Dino Mandrioli, *Fundamentals of Software Engineering*, 2nd Edition, Pearson, 2007.
7. Waman S Jawadekar, *Software Engineering-Principles and Practice*, Tata McGraw-Hill, 2004.

CO-PO Articulation Matrix Software Engineering Course (PC/CSE/9-T)

List of Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1. **Define** the various concepts related to software engineering. (LOTS: Level 1: Remember)	1	_	_	_	_	_	_	_	_	_	_	_	3	_	_
CO2. **Demonstrate** the use of stages of various Software Life Cycle Models. (LOTS: Level 2: Understanding)	1	_	_	_	_	_	_	_	_	_	_	_	3	_	_
CO3. **Apply** the Software Requirement Analysis and Software Design Process. (LOTS: Level 3: Apply)	2	1	1	_	2	_	_	_	_	_	2	_	3	_	_
CO4. **Analyse** the size, cost, complexity, reliability, quality and maintenance of a software system. (HOTS: Level 4: Analyse)	2	3	2	_	2	_	_	_	_	2	2	_	3	_	_
CO5. **Construct** software model according to the requirements of a customer. (HOTS: Level 6: Create)	3	3	2	_	3	_	_	_	2	2	3	_	3	_	_
Level of Attainments PC/CSE/9-T															

Java Programming

General Course Information

Course Code: PC/CSE/10-T Course Credits: 3 Type: Professional Core Contact Hours: 3 hours/week Mode: Lectures (L) Examination Duration: 3 Hours	**Course Assessment Methods (internal: 30; external: 70)** Two minor examinations (20 marks), Class Performance measured through percentage of lectures attended (4 marks), assignments (6 marks), and the end-semester examination (70 marks). For the end semester examination, nine questions are to be set by the examiner. A candidate is required to attempt 5 questions in all. All questions carry equal marks. Question number 1 will be compulsory and based on the entire syllabus. It will contain seven parts of 2 marks each. Question numbers 2 to 9 will be given by setting two questions from each of the four units of the syllabus. A candidate is required to attempt the remaining four questions by selecting one question from each of the four units.

Pre-requisites: The course assumes knowledge of Object-Oriented Concepts and programming in any Object-Oriented language.

About the Course:
Java is a general-purpose, concurrent, class-based, object-oriented computer programming language that is specifically designed to have as few implementation dependencies as possible. The aim of this course is to provide the students basic knowledge about object-oriented development and in-depth knowledge about syntax and programming techniques in Java.The course is very comprehensive and cover all the important Java concepts, e.g., Java basics, Object-Oriented Programming, Multithreading, File handling, Exception handling and more.

Course Outcomes: By the end of the course students will be able to:

CO1. **list** object-oriented characteristics peculiar to JAVA programming. (LOTS: Level 1: Remember)
CO2. **describe** object-oriented principles and paradigms implemented by Java programming language. (LOTS: Level 2: Understand)
CO3. **apply** object-oriented principles for solving problems using JAVA. (LOTS: Level 3: Apply)
CO4. **identify** classes, interfaces methods, hierarchy in the classes for a given programming problem in JAVA. (HOTS: Level 4: Analyse)
CO5. **design** Graphical User Interface applications and Web based applications in Java by importing applet, AWT and SWING packages. (HOTS: Level 6: Create)

Course Content

Unit I

Object-Oriented Programming Concepts: Object, Classes, Instantiation, Reuse, Procedural and object-oriented programming paradigms, Features of object-oriented programming: Encapsulation, Abstraction, Inheritance, and Polymorphism.
Java Programming Fundamentals: History of Java, Features of Java architecture, java architecture security, Garbage collections and Memory Management. Java programming language syntax, constants, variables, data types, operators, expressions. type conversion and casting. Control statements: if-else, for, while, & do-while loops and switch statements.

Methods, constructors, access specifiers and modifiers, Overloading methods and Overloading constructors. Recursion, building strings, exploring string class.

Unit II

Implementing OOP: Inheritance – Inheritance hierarchies, super and sub classes, super keyword, implementing inheritance, overriding methods, and interfaces**.** Implementing multiple inheritance using interfaces. Polymorphism- dynamic binding, Method Overriding, Abstract Methods and Classes.
Exception Handling: Exceptions in java, exception classes, built-in exceptions, try, catch and finally statements. Multiple catch statements, throw and throws statement. Creating and handling user-defined exceptions.
Working with Packages: Packages-Defining, Creating and Accessing a Package, Java API Packages, Using System Packages, Understanding CLASSPATH, importing packages.

Unit III

Working with GUI: Introduction to an Applet, life cycle of an applet, passing parameters to applets, applet security issues. AWT and Swing components, Layout Managers.
Event Handling: Delegation event model, event listeners, event handlers. Event classes: Action Event, Mouse Event. Event listeners interfaces: ActionListener, Mouse Listener, Mouse Motion Listener, adjustment Listener, Text Listener, Window Listener, Item Listener.
Threads: Multithreading, Thread class and Runnable interface. life cycle of a thread, Thread priority, thread synchronization.

Unit IV

File Handling: File class and Random-Access File class, Input and output streams, character streams, Object serialization, Serializable interface, Remote Method Invocation (RMI).
Database Connectivity: Introduction to SQL statements, ODBC/JDBC API: Connection, Statements, Prepared Statements, Record Set and execute statements.

Text and Reference Books:

1. Paul Deital, Harvey Deital, *Java™: How to Program*,9th Edition, Pearson Education (Prentice Hall), 2012.
2. Herbert Schildt, *Java™: The Complete Reference*,7th Edition, McGraw-Hill, 2007.
3. Kathy Sierra, Bert Bates, *Head First Java*, 2nd Edition, O'Reilly, 2005.
4. Ralph Bravaco , Shai Simoson, *Java Programming From the Ground Up*, Tata McGraw-Hill, 2009.
5. Sachin Malhotra, Saurabh Chaudhary, Programming in Java, Oxford University Press, 2011.
6. E. Balagurusamy, *Programming with Java: A Premier*, 3rd Edition, Tata McGraw-Hill, New Delhi, 2007.

CO-PO Articulation Matrix Java Programming Course (PC/CSE/10-T)

List of Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1. **List** object-oriented characteristics peculiar to JAVA programming. (LOTS: Level 1: Remember)	1	_	_	_	_	_	_	_	_	_	_	_	3	_	_
CO2. **Describe** object-oriented principles and paradigms implemented by Java programming language. (LOTS: Level 2: Understand)	1	_	_	_	_	_	_	_	_	_	_	_	3	_	_
CO3. **Apply** object-oriented principles for solving problems using JAVA. (LOTS: Level 3: Apply)	2	2	_	_	_	_	_	_	_	_	_	_	3	_	_
CO4. **identify** classes, interfaces methods, hierarchy in the classes for a given programming problem in JAVA. (HOTS: Level 4: Analyse)	2	3	3	_	2	_	_	_	_	_	_	_	3	2	_
CO5. **Design** Graphical User Interface applications and Web based applications in Java by importing applet, AWT and SWING packages. (HOTS: Level 6: Create)	3	3	3	_	2	_	_	_	2	_	_	_	3	_	_
Level of Attainments PC/CSE/10-T															

Microprocessors and Interfacing Lab.

General Information

Course Code: PC/CSE/5-P Course Credits: 1 Type: Professional Core Lab. Course Contact Hours: 2 hours/week Mode: Lab practice and assignments	**Course Assessment Methods (internal: 30; external: 70)** The internal assessment is based on the percentage of lab sessions attended (4 marks), timely submission of lab experiments/assignments and the quality of solutions provided in the assignments (16 marks), and an internal VIVA-VOCE (10 marks) conducted towards the end of semester. The external examination is of 70 marks. The break-up of marks for external examination is based on quality of lab reports (20 marks), quality of solution(s) for the given problem(s) at the time of examination (written work + execution of program(s)) (30) and VIVA-VOCE examination (20).

Pre-requisites: Basic concepts of Digital Electronics and Logic Design, Computer Organization

About the Course:
The objective of the lab course is to equip the students to design the Assembly Language programs for 8085 and 8086 microprocessors.

Course Outcomes: By the end of the course students will be able to:

CO1. **describe** the working of microprocessor kit/ TASM. (LOTS: Level 3: Apply)
CO2. **apply** interfacing of supporting chips with microprocessor. (LOTS: Level 3: Apply
CO3. **design** assembly language programs for the 8085 and 8086 microprocessors. (HOTS: Level 6: Create)
CO4. **analyse** the output of assembly language programs. (HOTS: Level 4: Analyse)
CO5. **create** lab records for the solutions of assignments. (HOTS: Level 6: Create)
CO6. **demonstrate** use of ethical practices, independent enquiry and team spirit. (LOTS: Level 3: Apply)

List of experiments/assignments

1. Two assignments to write assembly language programs using data transfer instructions
2. Two assignments to write assembly language programs using arithmetic instructions
3. Two assignments to write assembly language programs using flag manipulation instructions
4. Two assignments to write assembly language programs using shift and rotate instructions
5. Two assignments to write assembly language programs using stacks for 8086 micro-processor.
6. Two assignments to write assembly language programs using subroutines for 8086 micro-processor.
7. Two assignments on interfacing of supporting chips with 8085 and 8086 microprocessors.

Note:
The actual experiments/assignments will be designed by the course coordinator. One assignment should be designed to be done in groups of two or three students. The assignments must meet the objective of the course and the levels of the given course outcomes. The list of assignments and schedule of submission will be prepared by the course coordinator at the beginning of the semester.

CO-PO Articulation Matrix Micro-processor and Interfacing Lab. Course (PC/CSE/5-P)

List of Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1. **describe** the working of microprocessor kit/ TASM. (LOTS: Level 3: Apply)	2	-	-	-	1	-	-	-	-	-	-	1	2	-	-
CO2. **apply** interfacing of supporting chips with microprocessor. (LOTS: Level 3: Apply	2	-	-	-	1	-	-	-	-	-	-	1	3	-	-
CO3. **design** assembly language programs for the 8085 and 8086 microprocessors. (HOTS: Level 6: Create)	2	2	-	2	-	-	-	-	-	-	-	-	3	-	-
CO4. **analyse** the output of assembly language programs. (HOTS: Level 4: Analyse)	2	-	1	1	2	-	-	-	1	-	-	-	3	-	-
CO5. Create lab records for the solutions of assignments. (HOTS: Level 6: Create)	-	-	-	-	-	-	-	-	-	3	-	-	-	-	-
CO6. Demonstrate use of ethical practices, independent enquiry and team spirit. (LOTS: Level 3: Apply)	-	-	-	-	-	-	-	3	3	-	-	3	-	-	-
Level of Attainments PC/CSE/5-P															

Computer Networks Lab.

General Course Information

Course Code: PC/CSE/6-P Course Credits: 1 Type: Professional Core Lab. Course Contact Hours: 2 hours/week Mode: Lab. practice and assignments	**Course Assessment Methods (internal: 50; external: 50)** The internal and external assessment is based on the level of participation in lab. sessions and the timely submission of lab experiments/assignments, the quality of solutions designed for the assignments, the performance in VIVA-VOCE, the quality of lab. file and ethical practices followed. The internal examination is conducted by the course coordinator. The external examination is conducted by external examiner appointed by the Controller of Examination in association with the internal examiner appointed by the Chairperson of the Department.

Pre-requisites: knowledge of programming, digital and analog communication.

About the Course:

This course has been designed with an aim to provide students with an overview of the concepts and fundamentals of data communication and computer networks. Students learn about various topologies, network devices, routing protocols, firewall amongst other features and devices of Computer Networks.

Course Outcomes: By the end of the course students will be able to:

CO1. **demonstrate** various network topologies and networking devices. (LOTS: Level: 3: Apply)
CO2. **justify** a particular routing protocol for any implemented data communication networks. (HOTS: Level: 5: Evaluate)
CO3. **construct** a network and implement various network protocols. (HOTS: Level: 6: Create)
CO4. **devise** solutions for various routing and switching problems in Computer Networks. (HOTS: Level: 6: Create)
CO5. **create** lab records for the solutions of the assignments. (HOTS: Level: 6: Create)
CO6. **demonstrate** ethical practices, self-learning and team spirit. (LOTS: Level: 3: Apply)

List of Experiments/assignments:

1. a) Familiarization with networking components and devices: LAN Adapters - Hubs - Switches - Routers etc.
   b) Familiarization with transmission media and Tools: Co-axial cable - UTP Cable - Crimping Tool - Connectors etc.
2. Installation and introduction of simulation tools Packet Tracer/ GNS3.
3. Preparing the UTP cable for cross and direct connections using crimping tool.
4. Introduction to various interior and exterior routing protocols.
5. Configuration of RIP protocol on routers to configure a network topology.
6. Implementation EIGRP protocol on router.
7. Implementation OSPF protocol on a larger network.
8. Configuration of ARP protocol in network.
9. Configuration of a wireless device in simulated environment.
10. Implementation BGP protocol between two different networks.
11. Implementation of static routing in simulation environment.
12. Configuration of TELNET protocol on router for remote access.
13. Configuration of access lists on network to stop unwanted traffic on network.

14. Configuration of zone-based firewall in network.

Note:
The actual experiments/assignments will be designed by the course coordinator. One assignment should be designed to be done in groups of two or three students. The assignments must meet the objective of the course and the levels of the given course outcomes. The list of assignments and schedule of submission will be prepared by the course coordinator at the beginning of the semester.

CO-PO Articulation Matrix Computer Networks Lab. (PC/CSE/6-P)

List of Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO13	PSO1	PSO2	PSO3
CO1. **Demonstrate** various network topologies and networking devices. (LOTS: Level: 3: Apply)	1	1	_	_	_	_	_	_	_	_	_	_	_	3	_	_
CO2. **Justify** a particular routing protocol for any implemented data communication networks. (HOTS: Level: 5: Evaluate)	2	2	2	_	_	_	_	_	_	_	_	_	_	3	_	_
CO3. **Construct** a network and implement various network protocols. (HOTS: Level: 6: Create)	2	3	3	_	2	_	_	_	_	_	_	_	_	3	_	_
CO4. **Devise** solutions for various routing and switching problems in Computer Networks. (HOTS: Level: 6: Create)	3	3	3	3	3	_	_	_	_	_	_	_	_	3	_	_
CO5. **Create** lab records for the solutions of the assignments. (HOTS: Level: 6: Create)	_	_	_	_	_	_	_	_	_	3	--	--	_	_	_	_
CO6. **Demonstrate** ethical practices, self-learning and team spirit. (LOTS: Level: 3: Apply)	_	_	_	_	_	_	_	3	3	_	_	3	_	_	_	_
Level of Attainments PC/CSE/6-P																

Database Management System Lab.

General Course Information

Course Code: PC/CSE/7-P Course Credits: 1 Type: Professional Core Lab. Course Contact Hours: 2 hours/week Mode: Lab practice and assignments.	**Course Assessment Methods (internal: 50; external: 50)** The internal and external assessment is based on the level of participation in lab. sessions and the timely submission of lab experiments/assignments, the quality of solutions designed for the assignments, the performance in VIVA-VOCE, the quality of lab. file and ethical practices followed. The internal examination is conducted by the course coordinator. The external examination is conducted by external examiner appointed by the Controller of Examination in association with the internal examiner appointed by the Chairperson of the Department.

Pre-requisites: Exposure to a programming language, MS Access.

About the Course:
This lab. course on DBMS involves a rigorous training on Oracle programming. It provides a strong formal foundation in database concepts, technology and practice to the students to groom them into well-informed database application developers. The objective of the lab course is to develop proficiency in the execution of commands of the database design and query using Oracle.

Course Outcomes: By the end of the course students will be able to:

CO1. **implement** database problems using Oracle DML/DDL commands. (LOTS: Level 3: Apply)
CO2. **enforce** integrity constraints on a database using a state-of-the-art RDBMS. (LOTS: Level 3: Apply)
CO3. **analyse** the design of a relational database. (HOTS: Level 4: Analyse)
CO4. **design** a relational database for a given schema. (HOTS: Level 6: Create)
CO5. **create** lab assignment record that includes problem definitions, solutions, results and conclusions. (HOTS: Level 6: Create)
CO6. **demonstrate** ethical practices, self-learning and team spirit.

List of experiments/assignments:

1. Use oracle software and login with valid user id and password. Explore its GUI and practice some basic commands of it.
2. Three assignments related to creation of database with tables having different fields and datatypes.
3. Two assignments on the creation of table with different types of constraints.
4. Two assignments on insert, delete and modify records from the tables.
5. Two assignments on modifying the table using the alter command.
6. Two assignments on exploring select statement using various clauses like where, order by, group by, having and aggregate functions.
7. Two assignments on the use of set operations to query the tables.
8. Two assignments on creating joins and views on the tables.
9. One assignment on generating sub-queries.

Note:
The actual experiments/assignments will be designed by the course coordinator. One assignment should be designed to be done in groups of two or three students. The assignments

must meet the objective of the course and the levels of the given course outcomes. The list of assignments and schedule of submission will be prepared by the course coordinator at the beginning of the semester.

CO-PO Articulation Matrix Database Management System Lab. (PC/CSE/7-P)

List of Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1. Implement database problems using Oracle DML/DDL commands. (LOTS: Level 3: Apply)	2	1		_	2	_	_	_	_	_	_	_	3	_	_
CO2. **enforce** integrity constraints on a database using a state-of-the-art RDBMS. (LOTS: Level 3: Apply)	2	2	_	_	2	_	_	_	_	_	_	_	3	_	_
CO3. **Analyse** the design of a relational database. (HOTS: Level 4: Analyse)	3	3	1	_	2	_	_	_	_	_	_	_	3	_	_
CO4. **Design** a relational database for a given schema. (HOTS: Level 6: Create)	3	3	2	3	3	_	_	_	_	_	_	_	3	_	_
CO5. **Create** lab assignment record that includes problem definitions, solutions, results and conclusions. (HOTS: Level 6: Create)	_	_	_	_	_	_	_	_	_	3	_	_	_	_	_
CO6. Demonstrate ethical practices, self-learning and team spirit.	_	_	_	_	_	_	_	3	3	_	_	3	_	_	_
Level of Attainments PC/CSE/7-P															

Java Programming Lab.

General Course Information

Course Code: PC-CSE/10-P Course Credits: 2 Type: Professional Core Lab. Course Contact Hours: 4 hours/week Mode: Lab practice and assignments	**Course Assessment Methods (internal: 50; external: 50)** The internal and external assessment is based on the level of participation in lab. sessions and the timely submission of lab experiments/assignments, the quality of solutions designed for the assignments, the performance in VIVA-VOCE, the quality of lab. file and ethical practices followed. The internal examination is conducted by the course coordinator. The external examination is conducted by external examiner appointed by the Controller of Examination in association with the internal examiner appointed by the Chairperson of the Department.

Pre-requisites: The course assumes knowledge of Object-Oriented Concepts and programming.

About the Course:
This Java course will provide a strong understanding of basic Java programming elements and data abstraction using problem representation and the object-oriented framework. The objective of the lab course is to inculcate proficiency in students to design and develop market-based software applications.

Course Outcomes: By the end of the course students will be able to:

CO1. **implement** Java programs using object-oriented concepts for problem solving. (LOTS: Level 3: Apply)
CO2. **detect** syntax and logical errors in java programs (HOTS: Level 4: Analyse)
CO3. **apply** exception handling for making robust JAVA code. (HOTS: Level 3: Apply)
CO4. **design** java applications using File I/O and GUI. (HOTS: Level 6: Create)
CO5. **create** lab record of the solutions of assignments that includes problem definitions, solutions and conclusions. (HOTS: Level 6: Create)
CO6. **demonstrate** ethical practices, self-learning and team spirit. (LOTS: Level 3: Apply)

List of experiments/assignments:

1. Use eclipse or NetBeans platform and acquaint with the various menus, create a test project, add a test class and run it to see how you can use auto suggestions and auto fill functionalities. Try code formatter and code refactoring like renaming variables, methods and classes. Try debug step by step with a small program of about 10 to 15 lines which contains at least one if else condition and a for loop.
2. Two assignments illustrating class, objects, methods, arrays and various data types in java.
3. Two assignments on the use of control, looping statements and user defined functions.
4. One assignment illustrating the implementation of various forms of inheritance.
5. One assignment on method overloading.
6. One assignment on polymorphism and method overriding.
7. One assignment on implementing exception handling.
8. One assignment to illustrate interfaces in java.
9. One assignment to create package in java.
10. One assignment to design of multithreaded programs in java.
11. One new assignment on event handling.
12. Two assignments related to java applets.
13. One assignment to design a GUI application.

14. One assignment to access and update data from a database using JDBC.

Note:
The actual experiments/assignments will be designed by the course coordinator. One assignment should be designed to be done in groups of two or three students. The assignments must meet the objective of the course and the levels of the given course outcomes. The list of assignments and schedule of submission will be prepared by the course coordinator at the beginning of the semester.

CO-PO Articulation Matrix Java Programming Lab. (PC-CSE/10-P)

List of Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1. **Implement** Java programs using object-oriented concepts for problem solving. (LOTS: Level 3: Apply)	2	2	2	1	2	_	_	_	_	_	_	_	3	_	_
CO2. **Detect** syntax and logical errors in java programs (HOTS: Level 4: Analyse)	_	_	2	1	2	_	_	_	_	_	_	_	3	_	_
CO3. **Apply** exception handling for making robust JAVA code. (HOTS: Level 3: Apply)	2	2	1	1	_	_	_	_	_	_	_	_	3	_	_
CO4. **Design** java applications using File I/O and GUI. (HOTS: Level 6: Create)	3	3	3	1	3	_	_	_	_	3	_	_	3	_	_
CO5. **Create** lab record of the solutions of assignments that includes problem definitions, solutions and conclusions. (HOTS: Level 6: Create)	_	_	_	_	_	_	_	3	3	_	_	3	_	_	_
CO6. **Demonstrate** ethical practices, self-learning and team spirit. (LOTS: Level 3: Apply)	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_
Level of Attainments PC-CSE/10-P															