CURRICULUM

For

UNDERGRADUATE DEGREE COURSE IN

COMPUTER SCIENCE & ENGINEERING

(Engineering & Technology)

[Effective from 2018-19]

IIMT University, Meerut

SECOND YEAR, SEMESTER-III
STUDY & EVALUATION SCHEME

S. No.	Course Code	Course Name	Periods			Credit	Evaluation Scheme		
			L	T	P		External	Internal	Total
1	BTAS-301 / SOE-321/322/323/324/325/326/327	Mathematics-III / Science Based Open Elective	3	1	-	3	70	30	100
2	BTCS-301	Discrete Structure	3	1	-	3	70	30	100
3	BTCS-302	Data Structure Using C	3	1	-	3	70	30	100
4	BTCS-303	Object Oriented Programming with C++	3	1	-	3	70	30	100
5	BTEC-302	Digital Logic Design	3	1	-	3	70	30	100
6	BTCS-304	Computer Organization & Architecture	3	-	-	3	50	25	75
7	BTCS-352P	Data Structure using C Lab	-	-	2	2	50	25	75
8	BTCS-353P	Object Oriented Programming with C++ Lab	-	-	2	2	50	25	75
9	BTEC-352P	Digital Logic Design Lab	-	-	2	2	50	25	75
10	ECC-321	Industrial Visit/ Seminar/ Presentation on the report of visits						25	25
11	ECC-322	University Social Responsibility						25	25
12	ECC-323	Spoken Tutorial Certification				2		25	25
13	ECC-324	MOOC/ SWAYAM				2		25	25
		Total	**18**	**5**	**6**	**28**	**550**	**350**	**900**

Science Based Open Electives:

SOE-321 Laser System and Application
SOE-322 Manufacturing Process
SOE-323 Material Science
SOE-324 Nano Science
SOE-325 Nuclear Science
SOE-326 Polymer Science & Technology
SOE-327 Soft Computing

SECOND YEAR, SEMESTER-IV
STUDY & EVALUATION SCHEME

S. No.	Course Code	Course Name	Periods			Credit	Evaluation Scheme		
			L	T	P		External	Internal	Total
1	BTAS-401 / SOE-421/422/423/424/425/426/427	Mathematics-III / Science Based Open Elective	3	1	-	3	70	30	100
2	BTCS-401	Theory of Computation	3	1	-	3	70	30	100
3	BTCS-402	Operating System	3	1	-	3	70	30	100
4	BTCS-403	Object Oriented Programming with Java	3	1	-	3	70	30	100
5	BTCS-404	Software Engineering	3	1	-	3	70	30	100
6	BTHM-401	Engineering and Managerial Economics	3	-	-	3	50	25	75
7	BTCS-451P	UNIX & Shell Programming Lab	-	-	2	2	50	25	75
8	BTCS-452P	Operating System Lab	-	-	2	2	50	25	75
9	BTCS-453P	Object Oriented Programming with Java Lab	-	-	2	2	50	25	75
10	ECC-421	Industrial Visit/ Seminar/ Presentation on the report of visits						25	25
11	ECC-422	University Social Responsibility						25	25
12	ECC-423	Spoken Tutorial Certification				2		25	25
13	ECC-424	MOOC/ SWAYAM				2		25	25
		Total	**18**	**5**	**6**	**28**	**550**	**350**	**900**

Science Based Open Electives:

SOE-421 Laser System and Application
SOE-422 Manufacturing Process
SOE-423 Material Science
SOE-424 Nano Science
SOE-425 Nuclear Science
SOE-426 Polymer Science & Technology
SOE-427 Introduction to Soft Computing

Note:- Students will undergo for summer training of 4-6 weeks after 4th Sem.

SEMESTER-III

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

BTAS-301	MATHEMATICS-III	3L:1T:0P	3 credits

UNIT I **(L-8)**

Integral Transforms: Fourier integral, Fourier complex transform, Fourier sine and cosine transforms and applications to simple heat transfer equations. Z – Transform and its application to solve difference equations.

UNIT II **(L-8)**

Functions of a Complex Variable I: Analytic functions; C-R equations and harmonic functions; Line integral in the complex plane; Cauchy's integral theorem, Cauchy's integral formula for derivatives of analytic functions; Liouvilles theorem.

UNIT III **(L-8)**

Functions of a Complex Variable II: Representation of a function by power series; Taylor's and Laurent's series; Singularities, zeroes and poles; Residue theorem, evaluation of real integrals; conformal mapping and bilinear transformations.

UNIT IV **(L-8)**

Statistical Techniques: Moments, Moment generating functions, Skewness, Kurtosis, Curve Fitting and Solution of Equations: Method of lBTASt squares and curve fitting of straight line and parabola, Solution of cubic and bi-quadratic equations, Correlation and Regression, Binomial distribution, Poisson distribution, Normal distribution.

UNIT V **(L-8)**

Numerical Technique: Bisection method, Regula – Falsi method, Newton - Raphson method. **Interpolation:** Finite difference, Newton's forward and backward interpolation, Lagrange's and Newton's divided difference formula for unequal intervals; Numerical Differentiation, Numerical Integration; Trapezoidal, Simpson's 1/3 and 3/8 rule.

Course Outcome :

After completing the course, students should be able to:

1. Apply the fundamental concepts of integral transformation, complex variables and the numerical techniques for their resolution.
2. Solve the problems choosing the most suitable method.
3. Understand the difficulty of solving problems analytically and the need to use numerical approximations for their resolution.
4. Use computational tools to solve problems and applications of statistical techniques and numerical techniques.
5. Formulate and solve Fourier integral problems in the field of Industrial Organisation Engineering.
6. Use an adequate scientific language to formulate the basic concepts of the course.

Text Books

1. Das H.K., *Engineering Mathematics Vol-II*, S. Chand.
2. Grewal B.S., *Higher Engineering Engineering Mathematics*, Khanna Publishers.
3. Prasad C., *Engineering Mathematics for Engineers*, Prasad Mudralaya.
4 .Bali N.P., Engineering Engineering Mathematics-III, Laxmi Publications.
5. Kreyszig E., *Advanced Engineering Engineering Mathematics*, Wiley BTAStern.
7. Narayan Shanti, *A Text book of Matrices*, S. Chand.

Reference Books

1. Piskunov N, *Differential & Integral Calculus*, Moscow Peace Publishers

BTCS-301	DISCRETE STRUCTURE	3L:1T:0P	3 credits

UNIT-I **(L-8)**

Set Theory: Definition of Sets, Venn Diagrams, complements, cartesian products, power sets, counting principle, cardinality and countability (Countable and Uncountable sets), proofs of some general identitites on sets, pigeonhole principle.

Relation:Definition, types of relation, composition of relations, domain and range of a relation, pictorial representation of relation, properties of relation, partial ordering relation.

Function: Definition and types of function, composition of functions, recursively defined functions.

UNIT-II **(L-8)**

Propositional logic: Proposition logic, basic logic, logical connectives, truth tables, tautologies, contradiction , normal forms(conjunctive and disjunctive), modus ponens and modus tollens, validity, predicate logic, universal and existential quantification. Notion of proof: proof by implication, converse, inverse, contrapositive, negation, and contradiction, direct proof, proof by using truth table, proof by counter example.

UNIT-III **(L-8)**

Combinatories:Mathematical induction, recursive mathematical functions, permutations, combinations, inclusion-exclusion, recurrence relation and generating function

UNIT-IV **(L-8)**

Algebraic Structure:Binary composition and its properties, definition of algebraic structure; Semi group, Monoid Groups, Abelian Group, properties of groups, Permutation Groups, Sub Group, Cyclic Group, Ring and Field.

UNIT-V **(L-8)**

Graphs: Graph terminology, types of graph, connected graphs, components of graph, Euler graph, Hamiltonian path and circuits.
Tree: Definition, types of tree(rooted, binary), properties of trees, binary search tree, tree traversing (preorder, inorder, postorder).

Course Outcome :
After completing the course, students should be able to:

1. Write an argument using logical notation and determine if the argument is or is not valid.
2. Understand the basic principles of sets and operations in sets.
3. Prove basic set equalities.
4. Demonstrate an understanding of relations and functions and be able to determine their properties.
5. Determine when a function is 1-1 and "onto".
6. Demonstrate different traversal methods for trees and graphs.

Text Books

1. Hari Krishan, *Discrete Mathematics*, Pragati Edition

2. Lipchitz, S. & Lipson S., *Discrete Mathematics*, Outline series Tata McGraw Hill.
3. Kumar, S.S., *Discrete Mathematics,* S. Chand.
4. Deo, N., *Graph Theory with Applications to Engineering and Comp. Sci.*, Prentice Hall of India.

Reference books

1. Liu, C.L., *Elements of Discrete Mathematics*, McGraw Hil.
2. Dean, N., *Essence of Discrete Mathematics*, Prentice Hall.
3. Rosen, Kenneth H., *Discrete Mathematics and Its Applications*, McGraw Hill.
4. Johnsonbaugh, R., *Discrete Mathematics*, Macmillan.

BTCS-302	DATA STRUCTURE USING C	3L:1T:0P	3 credits

UNIT I **(L-8)**

Data Structure: Terminology, Operations, Elementary Data Organization, Algorithm Complexity and Time-Space trade-off.
Arrays: Definition, Representation and Analysis, Single and Multidimensional, address calculation, applications, Character String; String operation; Ordered List, Sparse Matrices.
Stacks: Array Representation and Implementation, Linked Representation, Operations; Push &Pop; Applications; Conversion of Infix to Prefix and Postfix Expressions, Evaluation of postfix expression

UNIT II **(L-8)**

Queues: Array Representation and Implementation, Linked Representation, Operations: Create, Add, Delete, Full and Empty, Types; Circular queue, Dequeue, Priority Queue;
Linked List: Representation and Implementation, Two-way Header List, Traversing and Searching, Overflow and Underflow, Operations; Insertion and deletion; doubly linked list, Garbage Collection and Compaction.

UNIT III **(L-8)**

Trees: Terminology, Binary Trees; Array and Linked Representation, Types: Complete, Extended. Threaded; Algebraic Expressions: Operations, Huffman algorithm. Tree Traversal algorithms: Inorder, Preorder and Postorder.

UNIT IV **(L-8)**

Sorting: Insertion, Bubble, Quick, Merge, and Heap Sort.
Binary Search Trees: Concepts, Insertion and Deletion in BST, Complexity of Search Algorithm, Path Length, AVL Trees, B-trees.

UNIT V **(L-8)**

Graphs: Terminology & Representations, Graphs vs. Multi- graphs, Directed, Representations, Adjacency Matrices, Traversal, Connected Component and Spanning Trees, Minimum Cost Spanning Trees, Shortest path algorithms: Dijkstra and Bellman ford.

Course Outcome :

After completing the course, students should be able to:

1 Ability to analyze algorithms and algorithm correctness.
2 Ability to summarize searching and sorting techniques
3 Ability to describe stack, queue and linked list operation
4 Ability to have knowledge of tree and graphs concepts.

Text Books

1. Lipschutz, S., *Data Structure*, Tata McGraw Hill.
2. Tenenbaum, A.M., *Data Structures using C & C++*, Prentice Hall of India.
3. Kanitkar, Y., *Data Structure using C++*, BPB.

Reference Books

1. Sahani, S.and Horowitz, E., *Fundamentals of Data Structures*, Galgotia
2. Kruse, R., *Data Structures and Program Design in C*, Pearson Education.
3. Cormen, T. H., *Introduction to Algorithms*, Prentice Hall of India.
4. Loudon, K., *Mastering Algorithms With C*,Shroff Publisher & Distributors.

BTCS-303	OBJECT ORIENTED PROGRAMMING WITH C++	3L:1T:0P	3 credits

UNIT I **(L-8)**

OOP: History, OOP vs. Procedure oriented programming, Abstraction, Encapsulation, Inheritance and Polymorphism. **Object& Classes:** Links and Associations, Generalization, Aggregation, Abstract classes, Metadata. **C++ Basics:** Structure of a program, Data types, Declaration of variables, Expressions, Operators, Operator Precedence, Evaluation of expressions, Type conversions, Pointers, Arrays, Pointers and Arrays, Strings, Structures. Flow control statement.

UNIT II **(L-8)**

Functions: Scope of variables; Parameter passing; Default arguments; Inline functions; Recursive functions; Pointers to functions. **Dynamic memory:** Allocation and Reallocation operators: new and delete; Preprocessor directives. **State model**: Events and States, Operations and Methods, Nested state diagrams, Concurrency, Relation of Object and Dynamic Models. **Functional Models:** Data flow diagrams, Specifying Operations, Constraints, OMT Methodologies, examples and case studies.

UNIT III **(L-8)**

C++ Classes and Data Abstraction: Definition, Structure, Objects, Scope, this pointer, Friends to a class, Static class members, Constant member functions, Constructors and Destructors, Data abstraction. **Polymorphism:** Function overloading; Operator overloading; Generic programming: necessity of templates, Function templates and class templates. **Inheritance:** Class hierarchy, Types, Base and Derived classes, Access to the base class members, Destructors, Virtual base class.

UNIT IV **(L-8)**

Virtual Functions and Polymorphism: Static and Dynamic bindings; Base and Derived class Virtual function: Definition, Call mechanism, pure virtual functions; Virtual destructors; Abstract classes; Implications of polymorphic use of classes.

UNIT V **(L-8)**

C++ I/O: I/O using C functions; Stream classes hierarchy; Stream I/O; File streams and String streams; Overloading << and >> operators; Error handling during file operations; Formatted I/O.

Course Outcome :

After completing the course, students should be able to:

1. Understand the features of C++ supporting object oriented programming
2. Understand the relative merits of C++ as an object oriented programming language
3. Understand how to produce object-oriented software using C++
4. Understand how to apply the major object-oriented concepts to implement object oriented programs in C++, encapsulation, inheritance and polymorphism

Text Books

1. Rambaugh, J., *Object Oriented Design and Modeling*, Prentice Hall of India.
2. Lafore,.R, *Object Oriented Programming in C+*, Galgotia.
3. Balagurusamy, E., *Object Oriented Programming with C++*, Tata McGraw Hill.

Reference Books

1. Lippman, S.B and Lajoie, J, *C++ Primer*, Pearson Education.
2. Stroutstrup, B., *The C++ Programming Language*, Pearson Education

BTEC-302	DIGITAL LOGIC DESIGN	3L:1T:0P	3 credits

UNIT-I **(L-8)**

Digital Systems: Binary Numbers, Octal, Hexa Decimal and other base numbers, Number base conversions, complements, signed binary numbers, Floating point number representation, binary codes, error detecting and correcting codes, digital logic gates(AND, NAND,OR,NOR, Ex-OR, Ex-NOR), Boolean algebra , basic theorems and properties, Boolean functions, canonical and standard forms.

UNIT-II **(L-8)**

Gate –Level Minimization and combination circuits, The K-Maps Methods, Three Variable, Four Variable, Five Variable , sum of products , product of sums Simplification, Don't care conditions , NAND and NOR implementation and other two level implantation.

UNIT-III **(L-8)**

Combinational Circuits (CC): Design Procedure, Combinational circuit for different code converters and other problems, Binary Adder, subtractor, Multiplier, Magnitude Comparator, Decoders, Encoders, Multiplexers, Demultiplexers.

UNIT-IV **(L-8)**

Synchronous Sequential Circuits: Latches, Flip-flops, analysis of clocked sequential circuits, design of counters, Up-down counters, Ripple counters , Registers, Shift registers, Synchronous Counters. Asynchronous Sequential Circuits: Reduction of state and follow tables, Role free Conditions.

UNIT-V **(L-8)**

Memory: Random Access memory, types of ROM, Memory decoding, address and data bus, Sequential Memory, Cache Memory, Programmable Logic Arrays, memory Hierarchy in terms of capacity and access time.

Course Outcome :

After completing the course, students should be able to:

1. Have a thorough understanding of the fundamental concepts and techniques used in digital electronics.
2. To understand and examine the structure of various number systems and its application in digital design.
3. The ability to understand, analyze and design various combinational and sequential circuits
4. The ability to identify and prevent various hazards and timing problems in a digital design.

5. To develop skill to build, and troubleshoot digital circuits.
6. Represent numbers and perform arithmetic in bases 2, 8, 10, and 16.
7. Simplify combinatorial circuits using Karnaugh maps.
8. Analyze and design modular combinatorial logic circuits containing decoders, multiplexers,

TEXT BOOKS:

1. Digital Design- M. Morris Mano.

REFERENCE BOOKS:

1. Switching and Finite Automata Theory by Zvi. Kohavi, Tata McGraw Hill.
2. Switching and Logic Design, C.V.S. Rao, Pearson Education.
3. Digital Principles and Design – Donald D.Givone, Tata McGraw Hill, Edition.
4. Fundamentals of Digital Logic & Micro Computer Design , 5TH Edition, M. Rafiquzzaman John Wiley.

BTCS-304	COMPUTER ORGANIZATION & ARCHITECTURE	3L:0T:0P	3 credits

UNIT I **(L-8)**

Introduction:. Digital computer generation, computer types and classifications, functional units and their interconnections, buses, bus architecture, types of buses and bus arbitration. Register and memory transfer

Central Processing Unit: Addition and subtraction of signed numbers, look ahead carry adders. Multiplication: Signed operand multiplication, Booths algorithm and array multiplier. Division and logic operations. Floating point arithmetic operation Processor organization, general register organization, stack organization and addressing modes.

UNIT II **(L-8)**

Comparison of Computer Organization & Architecture, Computer Components and Functions, Interconnection Structures. Bus Interconnections, Input / Output: I/O Module, Programmed I/O, Interrupt Driven I/O, Direct Memory Access

UNIT III **(L-8)**

System Organization Classification and design parameters, Memory Hierarchy, Internal Memory: RAM, SRAM and DRAM, Interleaved and Associative Memory. Cache Memory: Design Principles, Memory mappings, Replacement Algorithms, Cache performance, Cache Coherence. Virtual Memory, External Memory: Magnetic Discs, Optical Memory, Flash Memories, RAID Levels

UNIT-IV **(L—8)**

Processor Organization Instruction Formats, Instruction Sets, Addressing Modes, Addressing Modes Examples with Assembly Language [8085/8086 CPU] , Processor Organization, Structure and Function. Register Organization, Instruction Cycle, Instruction Pipelining. Introduction to RISC and CISC Architecture, Instruction Level Parallelism and Superscalar Processors: Design Issues.

Course Outcome :

After completing the course, students should be able to:

1. Master the binary and hexadecimal number systems including computer arithmetic,
2. Be familiar with the history and development of modern computers,
3. Be familiar with the Von Neumann architecture,
4. 4. Be familiar with the functional units of the processor such as the register file and arithmetic-logical unit,
5. Be familiar with the representation of data, addressing modes, instructions sets,

Text Books:

1. Ghosh T. K., "Computer Organization and Architecture", Tata McGraw-Hill, Third Edition, 2011.
2. William Stallings, "Computer Organization and Architecture – Designing for Performance", Pearson Education, Seventh Edition, 2006.

3. Behrooz Parahami, "Computer Architecture", Oxford University Press, Eighth Impression, 2011.
4. David A. Patterson and John L. Hennessy, "Computer Architecture-A Quantitative Approach", Elsevier, a division of reed India Private Limited, Fifth edition, 2012.
5. John P. Hayes, "Computer Architecture and Organization", Tata McGraw Hill, Third Edition, 1998.

BTCS-352P	DATA STRUCTURE USING C (LAB)	0L:0T:2P	2 credits

Note : Minimum eight practical's have to be performed

1. To write programs implementing Sorting programs: Bubble sort, Merge sort, Insertion sort, Selection sort, and Quick sort.
2. To write programs implementing Searching programs: Linear Search, Binary Search.
3. To write programs Array implementation of Stack, Queue, Circular Queue, and Linked List.
4. To write programs implementing Stack, Queue, and Circular Queue.
5. To write programs implementing Linked List using dynamic memory allocation.
6. To write program implementing Binary tree.
7. To write programs implementing Tree Traversals (pre-order, in-order, post-order).
8. To write programs implementing graph traversal (BFS, DFS).
9. To write programs implementing minimum cost spanning tree,
10. To write programs implementing single source shortest path Algorithm

Laboratory Outcome

At the end of this lab session, the student will

1. Be able to design and analyze the time and space efficiency of the data structure
2. Be capable to identity the appropriate data structure for given problem
3. Have practical knowledge on the applications of data structures
4. To implement various searching and sorting algorithm.

BTEC-352P	DIGITAL LOGIC DESIGN (LAB)	0L:0T:2P	2 credits

Note : Minimum eight experiments have to be performed

1. To study of various combinational circuits based on: AND/NAND Logic blocks and OR/NOR Logic blocks.
2. To study various waveforms at different points of a transistor bi-stable multi-vibrator and its frequency variation with different parameters.
3. Introduction to digital electronics lab- nomenclature of digital ICs, specifications, study of the data sheet, Concept of Vcc and ground, verification of the truth tables of logic gates using TTL ICs.
4. Implementation of the given Boolean function using logic gates in both SOP and POS forms.
5. Verification of state tables of RS, JK, T and D flip-flops using NAND & NOR gates
6. Implementation and verification of Decoder using logic gates.
7. Implementation and verification of Encoder using logic gates.
8. Implementation of 4:1 multiplexer using logic gates.
9. Implementation of 1:4 demultiplexer using logic gates.
10. Implementation of 4-bit parallel adder using 7483 IC.
11. Design, and verify the 4-bit synchronous counter.
12. Design, and verify the 4-bit asynchronous counter.

Laboratory Outcome:

At the end of this lab session, the student will

1. Describe and explain the operation of fundamental digital gates
2. Analyze the operation of medium complexity standard combinational circuits like the encoder, decoder, multiplexer, demultiplexer, adder
3. Analyze the operation of counters and shift registers
4. Design and operate practical digital logic circuits
5. Report findings and evaluate results
6. Create the appropriate truth table from a description of a combinational logic function.

BTCS-353P	OBJECT ORIENTED PROGRAMMING WITH C++ (LAB)	0L:0T:2P	2 credits

Note : Minimum eight experiments have to be performed

1. To write a program print Fibonacci series in C++.
2. To write a program for function overloading.
3. To write a program for using Function Overriding.
4. To write a program of Single level Inheritance.
5. To write a program of Multi level Inheritance.
6. To write a program of simple classes for understanding objects, member functions & constructors
    - classes with primitive data members,
    - classes with arrays as data members
    - classes with pointers as data members
    - classes with constant data members
    - classes with static member functions

7. To write a program illustrating how exception handling is done.

8. To write programs implementing various kinds of sorting algorithms, Search algorithms & Graph algorithms.

Laboratory Outcome

At the end of this lab session, the student will

1. Understand the process of writing, compiling and executing programs in C++
2. Using appropriate predefined functions in C++.
3. Implement the object oriented concepts in developing application using C++.
4. Developing applications in C++ using the understanding of Inheritance and polymorphism.
5. Understand and use exception handling while developing a C++ application.
6. To implement various searching and sorting algorithms are using C++.

SEMESTER-IV

BTCS-401	THEORY OF COMPUTATION	3L:1T:0P	3 credits

UNIT I **(L-8)**

Overview: Alphabets; Strings and Languages; Automata and Grammars; Finite automata (FA): Deterministic Finite Automata (DFA), Formal definition, simplified notations, Language of a DFA; NFA: Formal definition, Language of an NFA, Removing, Epsilon Transitions; Equivalence of DFAs and NFAs; Regular expressions (RE), Applications; Regular Grammars and FA, FA for Regular Grammar.

UNIT II **(L-8)**

Proving languages to be non-regular: Using Pumping Lemma; Closure Properties of Regular languages: Closure under Boolean operations, Reversal, Homomorphism, Inverse Homomorphism; Myhill-Nerode Theorem; DFA Minimization; Decision properties of Regular languages: Emptiness, Finiteness, Membership; Equivalence of Two DFAs or Res; Two-way finite automata; Finite automata with output.

UNIT III **(L-8)**

Context-free Grammars (CFGs): Definition; Sentential Forms; Language of a CFG; Derivation tree or Parse tree: Leftmost and Rightmost Derivations, Relationship between Parse Trees and Derivations, Parsing and Ambiguity; Ambiguity in Grammars and Languages; Pushdown Automata (PDA), Instantaneous Descriptions (Ids), Equivalence of acceptance by final state and empty stack; Equivalence of PDAs and CFGs, DPDAs and Regular Languages, DPDAs and CFLs.

UNIT IV **(L-8)**

Languages of DPDAs: DPDAs and ambiguous grammars; Simplification of CFGs: Normal forms; CNF and GNF Proving that some languages are not context free: Pumping lemma for CFLs; Closure Properties of CFLs: Closure Under Union, Concatenation, Kleene Closure, Substitution, Reversal, Intersection with Regular Set; Decision Properties of CFLs.

UNIT V **(L-8)**

Turing Machines (TM): Behavior, Transition Diagrams, Language of a TM; TM: Accepters, Deciders, Computer of integer functions; Programming techniques for TMs: Storage in state, Multiple Tracks,Variants of TMs: Multi tape TMs, Nondeterministic TMs, TMs with Semi-Infinite Tapes, Multi Stack Machines; Un-decidability of the universal language; The Halting problem; Un-decidable problems about TMs, Post's Correspondence Problem (PCP): Definition, Un-decidability of PCP; Other Un-Decidability Problems, Problems related to CFLs; Context Sensitive Language and Linear Bounded Automata; Chomsky Hierarchy.

Course Outcome :

After completing the course, students should be able to:

1. Students will demonstrate knowledge of basic mathematical models of computation and describe how they relate to formal languages.

2. Students will understand that there are limitations on what computers can do, and learn examples of unsolvable problems.
3. Students will learn that certain problems do not admit efficient algorithms, and identify such problems

Text Books

1. Ullman, H., *Introduction to Automata Theory, Language and Computation*, Narosa Publishing House.
2. Mishra, K.L.P. and Chandrasekaran, N., *Theory of Computer Science (Automat Language and Computation)*, Prentice Hall of India.
3. Martin, J.C., *Introduction to Language and Theory of Computation*, Tata McGraw Hill.

Reference Books

1. Papadimitrou, C. & Lewis, *Elements of Theory of Computation*, Prentice Hall of India.
2. Cohen, D., *Introduction to computer Theory*, John Wiley& Sons.

BTCS-402	OPERATING SYSTEM	3L:1T:0P	3 credits

UNIT I **(L-8)**

Operating System: History, Types: Batch System, Time Sharing System, Real Time System, Multiprogramming, Distributed System; Functions; Services; System calls; System programs; Virtual machines.

UNIT II **(L-8)**

Process Management: Concept, States, Control Block, Scheduling; CPU, Criteria, Algorithms, Preemptive& Non Preemptive.

UNIT III **(L-8)**

Process Synchronization: Critical Section, Race Condition, Synchronization Hardware, Semaphores, Classical Problems of Synchronization.

Deadlocks: Characterization, Avoidance, Detection & Recovery.

UNIT IV **(L-8)**

Memory Management: Contiguous Allocation, External and Internal Fragmentation, Paging &Segmentation.

Virtual Memory: Concept, Demand Paging, Page Replacement Algorithms, Allocation of Frames, Thrashing.

UNIT V **(L-8)**

File Management: Directory Structure, Allocation Methods; Contiguous; Linked; Indexed: Free Space Management; Disk: Structure, Scheduling Algorithms, Management.

Course Outcome :

After completing the course, students should be able to:

1. Demonstrate understanding of the concepts, structure and design of operating Systems
2. Demonstrate understanding of operating system design and its impact on application
3. System design and performance Demonstrate competence in recognizing and using operating system features.
4. understand and analyse theory and implementation of: processes, resource control (concurrency etc.), physical and virtual memory, scheduling, I/O and file

Text Books

1. Silbershatz, A. and Galvin, P., *Operating System Concept*, Addison-Wesley.
2. Nutt, G., *Operating Systems*, Addison-Wesley.
3. God bole, A., *Operating System*, Prentice Hall of India.

References Books

1. Flynn, M., *Understanding Operating System*, Thomson Press.
2. Tannenbaum, O., *Operating System Concept*, Addison-Wesley.
3. Joshi, R.C., and Tapaswi, S., *Operating Systems*, Wiley-Dreamtech.

BTCS-403	OBJECT ORIENTED PROGRAMMING WITH JAVA	3L:1T:0P	3 credits

UNIT-I OBJECT ORIENTED PROGRAMMING : (L-8)

Introduction to OOP, Objects and Classes, Characteristics of OOP, Difference between OOP and Procedure Oriented Programming, Summary

Introduction to Java Programming : Introduction, Features of Java, Comparing Java and other languages, Applications and Applets, Java Development Kit, More Complex Programs, Java Source file structure, Prerequisites for Compiling and Running Java Programs.

Java Language Fundamentals : The building Blocks of Java – Data types – variable declarations – wrapper classes – Operators and Assignment – Control structures – Arrays – Strings – The String Buffer Class.

UNIT-II JAVA AS AN OOP LANGUAGE : (L-8)

Defining classes – Modifiers – Packages – Interfaces.

Exception Handling : Introduction – Basics of Exception Handling in Java – Exception Hierarchy – Constructors and Methods in Throwable class – Unchecked and checked exceptions – Handling exceptions in Java – Exception and Inheritance – Throwing User defined Exceptions – Redirecting and Rethrowing Exceptions – Advantages of Exception – Handling Mechanism.

UNIT-III MULTITHREADING : (L-8)

Introduction : An Overview of Threads – Creating Threads – Thread Life – cycle – Thread priorities and Thread scheduling – Thread synchronization – Thread groups – Communication of Threads.

Files and I/O Streams : An Overview of I/O streams – Java I/O – File streams – File Input stream and File output stream – Filter streams – Random Access File – Serialization.

UNIT-IV DATABASE HANDLING USING JDBC : (L-8)

An Overview of DBMS – JDBC Architecture – Working with JDBC

Servlets : Introduction – How to run servlets – The Life – cycle of the servlet – servlet API – Multitier Applications using JDBC from a servlet.

UNIT-V APPLETS AND RMI: (L-8)

Applets : Introduction – Java applications versus Java Applets – Applet Life cycle – Working with Applets – The HTML Applet Tag.

Networking and Remote Method Invocation : Introduction to Networking – Understanding Ports – Networking classes in JDK – Introduction to RMI – RMI Architecture – Implementing Remote class and interface – security.

Course Outcome :

After completing the course, students should be able to:

1. Explain the many concepts in Object-oriented programming paradigm; understand the history behind the Java technology, its features and strengths;
2. understand and explain how Java achieves platform portability;
3. understand the theory behind many programming structures, constructs, library codes
4. exposed by the Java language; knowledge of the structure and model of the Java programming language, (knowledge)

Text Books:

1. James Rumbaugh etal, "Object Oriented Modeling and Design", PHI
2. Herbert Schieldt, "The Complete Reference: Java", TMH.
3. E. Balagurusamy, "Programming in JAVA", TMH.

References:

1. Booch Grady, "Object Oriented Analysis & Design with application 3/e", Pearson Education, New Delhi.
2. Bjarne Stroustrup, "C++ Programming Language", Addison Wesley 3. E. Balagurusamy, "Object Oriented Programming with C++", TMH.

BTCS-404	SOFTWARE ENGINEERING	3L:1T:0P	3 credits

UNIT I **(L-8)**

Introduction: Evolution and impact of Software Engineering, Software Development Life Cycle (SDLC) Models: Waterfall Model, Prototype Model, Spiral Model, Agile methodology, Layered Approach. **Software Requirements Analysis and Specifications** FBTASibility Study, Functional and Non-Functional Requirements, Requirements Gathering, Requirement Analysis and Specifications using DFD, Data Dictionaries and ER Diagrams, Requirements documentation, Characteristics and Organization of Software Requirement Specifications (SRS)

UNIT II **(L-8)**

Software-Design and Coding: Principles; Problem Partitioning; Abstraction; Top-Down and Bottom-Up design; Structured Approach; Functional vs. Object Oriented Approach; UML, Design Specifications and Verification; Cohesion; Coupling. Distributed Software Design, User Interface Design, Coding standards and Code Review Techniques

UNIT III **(L-8)**

Software Testing : Software Testing Fundamentals, SDLC Testing : Unit Testing, Integration Testing, System Testing, Regression Testing, Smoke Testing, Security Test, Stress Test, Performance Test, Functional Testing or Black Box Testing: Boundary Value Analysis, Alpha Testing, Beta Testing, and Acceptance Testing, Structural Testing or White Box Testing: Basis Path Testing, DD-Paths, Cyclomatic Complexity, Data Flow Testing, Mutation.

UNIT IV **(L-8)**

Test Management: Test Cycle, Test Estimation, Test Cases, Test Scenarios **Testing Tools:** Static, Dynamic, Characteristics of Modern Tools and Automation.

UNIT V **(L-8)**

Software Maintenance: Updates-Upgrades-Patches-Versions, Error Reporting, Customer Support, Maintenance Process, **Software Reliability**: Importance, Hardware Reliability and Software Reliability, Failure and Faults, Reliability Models, Software Reuse,

Course Outcome :

After completing the course, students should be able to:

1. Correctly create a model of the structure and behavior of a software system.
2. Design and implement, in a programming language, an executable solution to a given problem using common software principles and best practices.
3. Apply appropriate software testing techniques and evaluate the quality of a software product at module, integration, and system granularity levels.

4. Select and adapt suitable elements from among conventional and evolving software development life-cycle processes and apply the resulting process to a software project.
5. Collaborate in teams to develop a significantly sized software system from conceptualization to completion.
6. Apply new software models, techniques and technologies to bring out innovative and novelistic solutions for the growth of the society in all aspects and evolving into their continuous professional development.

Text Books

1. Agarwal, K.K., *Software Engineering*, New Age International.
2. Pressman, R.S., *Software Engineering: A Practitioner's Approach*, McGraw Hill.
3. Jalote, P., *Software Engineering*, Narosa Publishing House.
4. Perry, W., Effective Methods for Software Testing, John Wiley & Sons.
5. Tamres, L., Software Testing, Pearson Education.
6. Robert, V. B., Testing Object-Oriented Systems-Models, Patterns and Tools, Addison-Wesley.

Reference Books

1. Sommerville, I., *Software Engineering*, Addison-Wesley.
2. Aggarwal, K.K. & Singh, Y., *Software Engineering*, New Age International Publishers.
3. Boris, B., *Software Testing Techniques*, Van Nostrand Reinhold.

BTHM-401	ENGINEERING AND MANAGERIAL ECONOMICS	3L:0T:0P	3 credits

UNIT I **(L-8)**

Introduction: Meaning, Nature and Scope of Economics, Meaning of Science, Engineering and Technology, Managerial Economics and its scope in engineering perspective.

UNIT-II **(L-8)**

Demand: Basic Concepts Demand Analysis, Law of Demand, Determinates of Demand, Elasticity of Demand-Price, Income and cross Elasticity, Uses of concept of elasticity of demand in managerial Decision.

UNIT-III **(L-8)**

Forecasting: Demand forecasting Meaning, significance and methods of demand forecasting, production function, Laws of returns to scale & Law of Diminishing returns scale.

Short and Long run Cost curves: fixed cost, variable cost, average cost, marginal cost, Opportunity cost.

UNIT-IV **(L-8)**

Inflation: National Income, Inflation and Business Cycles Concept of N.I. and MBTA Surement, Meaning of Inflation, Type causes & prevention methods, Phases of business cycle.

Market Study: Market Structure Perfect Competition, Imperfect competition: Monopolistic, Oligopoly, Duopoly sorbent features of price determination and various market conditions

Course Outcome :

After completing the course, students should be able to:

1. Evaluate the economic theories, cost concepts and pricing policies
2. Understand the market structures and integration concepts
3. Apply the concepts of financial management for project appraisal
4. Understand the impact of inflation, Financial planning, economic basis for replacement, project scheduling, and legal and regulatory issues are introduced and applied to economic investment and project-management problems.

Text Books

1. Dwivedi, D.N., *Managerial Economics*, Vikas Publishing.
2. Maheshwari, Y., *Managerial Economics*, Prentice Hall of India.

Reference Books

1. Koutsoyiannis, *A : Modern Microeconomics*, ELBS.
2. Kakkar, D.N., *Managerial Economics for Engineering*, New Age International publication.

BTCS-451P	UNIX & SHELL PROGRAMMING (LAB)	0L:0T:2P	2 credits

Note : Minimum six experiments have to be performed

1. To write Shell Script for UNIX environment.
2. To implement basic commands of UNIX
3. To implement commands of UNIX administration, user authorization, grant of users right and privileges, backup and recovery.
4. To study Source Code Control System understanding LEX and YACC, debugger tools (Lint, make etc.)
5. To write program in C for Process Creation, Parent/Child process relationship, forking of process.
6. To write program for Inter Process Communication
7. To write program for socket programming implementation of exec system call, pipe.
8. To write program for semaphore and message queue.

Laboratory Outcome

At the end of this lab session, the student will

1. You will be able to run various UNIX commands on a standard UNIX/LINUX Operating system
2. You will be able to run C / C++ programs on UNIX.
3. You will be able to do shell programming on UNIX OS
4. You will be able to understand and handle UNIX system calls.

BTCS-452P	OPERATING SYSTEM (LAB)	0L:0T:2P	2 credits

Note : Minimum eight experiments have to be performed

1. Write a C program to simulate the following non-preemptive CPU scheduling algorithms to find turnaround time and waiting time. a) FCFS b) SJF c) Round Robin d) Priority
2. Write a C program to simulate the following preemptive CPU scheduling algorithms to find turnaround time and waiting time. a) FCFS b) SJF c) Round Robin d) Priority
3. Write a C program to simulate the following contiguous memory allocation techniques a) Worst-fit b) Best-fit c) First-fit
4. Write a C program to simulate paging technique of memory management.
5. Write a C program to simulate Bankers algorithm for the purpose of deadlock avoidance
6. Write a C program to simulate disk scheduling algorithms a) FCFS b) SCAN c) C-SCAN
7. Write a C program to simulate page replacement algorithms a) FIFO b) LRU c) LFU
8. Write a C program to simulate producer-consumer problem using semaphores.
9. Write a C program to simulate the concept of Dining-Philosophers problem.
10. Write a C program to simulate the following file allocation strategies. a) Sequential b) Linked c)) Indexed

Laboratory Outcome

At the end of this lab session, the student will

1. To make students able to implement CPU scheduling algorithms and Bankers algorithm used for deadlock avoidance and prevention.
2. Students will also be able to implement page replacement and memory management algorithms.
3. To simulate CPU Scheduling Algorithms like FCFS, Round Robin, SJF, and Priority.
4. .To Implement memory management schemes and page replacement schemes.
5. To Simulate file allocation and organization techniques.
6. To Understand the concepts of deadlock in operating systems and implement them in multiprogramming system.

BTCS-453P	OBJECT ORIENTED PROGRAMMING WITH JAVA LAB	0L:0T:2P	2 credits

Note : Minimum ten experiments have to be performed

1. To write a java program to find volume of box
2. To write a java program to handle string using string function.
3. To write a java program to find the account balance using package
4. To write a java program using user defined interface concept.
5. To write a java program using pre defined interface concept.
6. To implement multi threading and exception handling using java program
7. To write a java program to implement possible operations on matrix.
8. To write a java program to sorting the characters of a given string
9. To write a java program to implement the pre defined exception concept in java
10. To write a java program to implement the user defined exception concept in java
11. To write a java program to implement authentication technique using applets.
12. To write a java program to implement event handling in Applets.
13. To write a java program to Program for Deadlock Handling.
14. To write a java program to Program for Producer – Consumer problem.
15. To write a java program to implement Process Synchronization .

Laboratory Outcome

At the end of this lab session, the student will

1. Able to use Java compiler and eclipse platform to write and execute java program.
2. Implement Object Oriented Programming Concepts(class, constructor, overloading, inheritance, overriding) in java
3. Understand and Apply Object oriented features and Java concepts.
4. Able to access data from a Database with java program.
5. Develop applications using Console I/O and File I/O,GUI applications
6. Use and create packages and interfaces in a Java program
7. Create Applets
8. Implements exception handling in Java.
9. Implement Multithreading in java.
10. Use of Input/output Streams in java