



**Janardan Bhagat Shikshan Prasarak Sanstha's
CHANGU KANA THAKUR
ARTS, COMMERCE & SCIENCE COLLEGE,
NEW PANVEL (AUTONOMOUS)**

**Re-accredited 'A+' Grade by NAAC
'College with Potential for Excellence' Status Awarded by UGC
'Best College Award' by University of Mumbai**

Program: B.Sc

**Revised Syllabus of F.Y.B.Sc. Information Technology
Choice Based Credit & Grading System (60:40)
w.e.f. Academic Year 2019-2020**

| Sr. No. | Heading | Particulars |
|---------|---------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | Title of Course | Information Technology |
| 2 | Eligibility for Admission | <p>(a) A candidate for being eligible for admission to the degree course of Bachelor of Science-Information Technology, shall have passed XII standard examination of the Maharashtra Board of Higher Secondary Education or it's equivalent with Mathematic and Statistics as one of the subject and should have secured not less than 45% marks in aggregate for open category and 40% marks in aggregate in case of Reserved category candidates.</p> <p>(b) Candidate who have passed Diploma (Three years after S.S.C. – Xth Std.) in Information Technology/ Computer Technology/ Computer Engineering/Computer Science/ Electrical, Electronics and Video Engineering and Allied Branches/Mechanical and Allied Branches/ Civil and Allied branches are eligible for direct admission to the Second Year of the B.Sc. (I.T.) degree course.</p> <p>(c) However, the Diploma should be recognized by the Board of Technical Education or any other recognized Government Body. Minimum marks required 45% aggregate for open category candidates and 40% aggregate for reserved category candidates.</p> <p>OR</p> <p>Candidates with post HSC-Diploma in Information Technology/Computer Technology/ Computer Engineering/ Computer Science/ and Allied branches will be eligible for direct admission to the Second Year of B.Sc. (I.T.). However, the Diploma should be recognized by the Board of Technical Education or any other recognized Government Body Minimum Marks required 45% aggregate for open category candidates and 40% aggregate for reserved category candidates.</p> |
| 3 | Passing marks | 40% |

F.Y.B.Sc, Information Technology Syllabus

| | | |
|---|--------------------------------------|------------------|
| 4 | Ordinances/Regulations (if any) | |
| 5 | No. of Semesters | Two |
| 6 | Level | U.G. |
| 7 | Pattern | Semester (60:40) |
| 8 | Status | Revised |
| 9 | To be implemented from Academic year | 2019-2020 |

Preamble of the Syllabus:

The B.Sc. Information Technology programme is started with an aim to make the learners employable and impart industry oriented training.

The main objectives of the course are:

- To think analytically, creatively and critically in developing robust, extensible and highly maintainable technological solutions to simple and complex problems.
- To apply their knowledge and skills to be employed and excel in IT professional careers and/or to continue their education in IT and/or related post graduate programmes.
- To be capable of managing complex IT projects with consideration of the human, financial and environmental factors.
- To work effectively as a part of a team to achieve a common stated goal.
- To adhere to the highest standards of ethics, including relevant industry and organizational codes of conduct.
- To communicate effectively with a range of audiences both technical and non-technical.
- To develop an aptitude to engage in continuing professional development.

Objectives of the Course:

The new syllabus is aimed to achieve the objectives. The syllabus spanning three years covers the industry relevant courses. The students will be ready for the jobs available in different fields like:

- Software Development (Programming)
- Website Development
- Mobile app development
- Embedded Systems Programming
- Embedded Systems Development
- Software Testing
- Networking
- Database Administration
- System Administration
- Cyber Law Consultant
- GIS (Geographic Information Systems)
- IT Service Desk
- Security

Course Outcome: By the end of the course, a student should develop the Ability:

- Learners are able to use and apply current technical concepts and practices in the core information technologies.
- Learners are able to apply knowledge of computing and mathematics appropriate to the discipline.
- Learners are able to analyze a problem, and identify and define the computing requirements appropriate to its solution.
- Learners are able to effectively integrate IT based solutions into the user environment.
- Learners are able to design, implement, and evaluate a computer based system, process, component, or program to meet desired needs.

Scheme of Examination

A) Internal Assessment: 40 %

40 Marks

| Sr. No. | Particular | Marks | |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|----------|
| 01 | One periodical class test / online examination to be conducted in the given semester | 20 Marks | |
| 02 | One case study/ project with presentation based on curriculum to be assessed by the teacher concerned | 15 Marks | |
| | Presentation | | 10 Marks |
| | Written Document | | 05 Marks |
| 03 | Active participation in routine class instructional deliveries and overall conduct as a responsible learner, mannerism and articulation and exhibit of leadership qualities in organizing related academic activities | 05 Marks | |

Question Paper Pattern for Class Test

| Question No. | Particular | Marks |
|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|--------------|
| Q-1 | Match the Column / Fill in the Blanks / Multiple Choice Questions/ Answer in One or Two Lines (Concept based Questions) (1 Marks / 2 Marks each) | 10 Marks |
| Q-2 | Answer in Brief (Attempt any Two of the Three) (5 Marks each) | 10 Marks |

B) Semester End Examination: 60 %

60 Marks

- Duration: The examination shall be of 2 hours duration.

| Theory question paper pattern |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1. There shall be five questions each of 12 marks. 2. All questions shall be compulsory with internal options. 3. Question may be subdivided into sub-questions a, b, c... and the allocation of marks depends on the weightage of the unit. |

➤ Passing Standard

The learners to pass a course shall have to obtain a minimum of 40% marks in aggregate for each course where the course consists of Internal Assessment and Semester End Examination. The learners shall obtain minimum of 40% marks (i.e. 16 out of 40) in the Internal Assessment and 40% marks in Semester End Examination (i.e. 24 Out of 60) separately, to pass the course and minimum of Grade D, wherever applicable, to pass a particular semester. A learner will be said to have passed the course if the learner passes the Internal Assessment and Semester End Examination together.

Choice Based Credit Grading and Semester System (CBCGS)
F.Y.B. Sc. Information Technology Syllabus
To be implemented from the Academic year 2019-2020
SEMESTER I

| Course Code | Course Type | Course Title | Credits |
|----------------------|--------------------------------------------|-------------------------------------------|----------------|
| UIT1ICP | Core Subject | Introduction to C++ programming | 2 |
| UIT1DEL | Core Subject | Digital Electronics | 2 |
| UIT1OPS | Core Subject | Operating Systems | 2 |
| UIT1DMS | Core Subject | Discrete Mathematics | 2 |
| UIT1CMS | Ability Enhancement Skill Course | Communication Skills | 2 |
| UIT1CPP | Core Subject Practical | Introduction to C++ Programming Practical | 2 |
| UIT1DEP | Core Subject Practical | Digital Electronics Practical | 2 |
| UIT1OSP | Core Subject Practical | Operating Systems Practical | 2 |
| UIT1DMP | Core Subject Practical | Discrete Mathematics Practical | 2 |
| UIT1CSP | Ability Enhancement Skill Course Practical | Communication Skills Practical | 2 |
| Total Credits | | | 20 |

Choice Based Credit Grading and Semester System (CBCGS)
F.Y.B. Sc. Information Technology Syllabus
To be implemented from the Academic year 2019-2020
SEMESTER II

| Course Code | Course Type | Course Title | Credits |
|----------------------|--------------------------------------------|---------------------------------------|----------------|
| UIT2OOP | Core Subject | Object Oriented Programming | 2 |
| UIT2MPA | Core Subject | Microprocessor Architecture | 2 |
| UIT2DMS | Core Subject | Database Management System | 2 |
| UIT2NMS | Core Subject | Numerical Methods | 2 |
| UIT2WPR | Ability Enhancement Skill Course | Web Programming | 2 |
| UIT2OPP | Core Subject Practical | Object Oriented Programming Practical | 2 |
| UIT2MAP | Core Subject Practical | Microprocessor Architecture Practical | 2 |
| UIT2DSP | Core Subject Practical | Database Management System Practical | 2 |
| UIT2NMP | Core Subject Practical | Numerical Methods Practical | 2 |
| UIT2WRP | Ability Enhancement Skill Course Practical | Web Programming Practical | 2 |
| Total Credits | | | 20 |

Semester I

| | | |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| Course Code UIT1ICP | Introduction to C++ Programming | |
| Objectives The objective of this course is to introduce the concept of the basic programming language with C++. | | |
| Expected Learning Outcomes: 1) Learners should be able to understand how C++ improves C with object-oriented features. 2) Learners should be able to learn how to write inline functions for efficiency and performance 3) Learners should be able to write programs that are very efficient in memory usage. | | |
| I | Programming Logic and techniques : Algorithms, Flow-charts, Program Design, Introduction to C++: Origin of C++, A Sample C++ program, pitfall and programming tips. Testing and Debugging, Variables and Assignments: variables, identifiers, variable declarations, Assignment Statements, reference variable, symbolic constant. Input and Output: cin, cout, escape sequences, include directives and Namespaces, Indenting and Comments, Operator precedence, Data types and Expressions, Arithmetic operators, Type compatibilities. | 12 |
| II | Flow of Control : Compound statements, Loops: while, for, do while, nested loops, Decision making: if – else, nested if else, switch , break and continue, Manipulators: endl, setw, sizeof, Increment and decrement operators, Type Cast Operators, Scope resolution operators | 12 |
| III | Functions: Function Prototypes, built in functions and user defined functions, Function overloading, Call by reference, Call by value, const member functions. Inline Functions and recursive functions, Math Library Functions. | 12 |
| IV | Derived Data types (Arrays , pointers , functions): Introduction to arrays, arrays in functions, 2-D arrays , Multidimensional arrays, Introduction to pointers, void pointers, pointers in function, pointer to constant and constant pointer, generic pointer. | 12 |
| V | Strings and Vectors: String functions: strcmp, strcat, strlen, strcpy. Vector Basics. Introduction to Structures: Structure Variables, Initialization, Structure Assignment, Nested Structure. Structures and Functions, Structures and Arrays: Arrays of Structures, Structures Containing Arrays, Unions. | 12 |

| Course Code | Practical List |
|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| UIT1CPP | <p>1. Basic Programs:</p> <p>a. Write a program to display the message HELLO WORLD.</p> <p>b. Write a program to declare some variables of type int, float and double. Assign some values to these variables & display these values.</p> <p>c. Write a program to find the addition, subtraction, multiplication and division of two numbers.</p> <p>2. Programs on variables:</p> <p>a. Write a program to swap two numbers without using third variable.</p> <p>b. Write a program to find the area of rectangle, square and circle.</p> <p>c. Write a program to find the volume of a cube, sphere, and cylinder.</p> <p>3. Conditional statements and loops(basic)</p> <p>a. Write a program to enter a number from the user and display the month name. If number>13 then display invalid input using switch case.</p> |

- b. Write a program to check whether the number is even or odd.
- c. Write a program to check whether the number is positive, negative or zero.
- d. Write a program to find the factorial of a number.
- e. Write a program to check whether the entered number is prime or not.
- f. Write a program to find the largest of three numbers.

4. Conditional statements and loops(advanced)

- a. Write a program to find the sum of squares of digits of a number.
- b. Write a program to reverse the digits of an integer.
- c. Write a program to find the sum of numbers from 1 to 100.
- d. Write a program to print the Fibonacci series.
- e. Write a program to find the reverse of a number.
- f. Write a program to find whether a given number is palindrome or not.
- g. Write a program that solve the quadratic equation
- h. Write a program to check whether the entered number is Armstrong or not.
- i. Write a program to count the digit in a number

5. Programs on patterns:

- a. Programs on different patterns.

6. Functions:

- a. Programs on Functions.

7. Recursive functions

- a. Write a program to find the factorial of a number using recursive function.
- b. Write a program to find the sum of natural number using recursive function.

8. Arrays

- a. Write a program to find the largest value that is stored in the array.
- b. Write a program to compute the sum of all elements stored in an array.
- c. Write a program to arrange the 'n' numbers stored in the array in ascending and descending order.
- d. Write a program that performs addition and subtraction of matrices.
- e. Write a program that performs multiplication of matrices.

9. String handling

- a. String operations for string length , string concatenation
- b. String operations for string reverse, string comparison,
- c. Console formatting functions.

10. Structures and Unions

- a. Programs on structures.
- b. Programs on unions

Reference Books:

- 1) "Let us C++" , Y.P.Kanetkar, Seventh edition, BPB publication
- 2) "Problem Solving with C++" , Walter Savitch, Sixth Edition, Pearson Education.
- 3) Schaum's outlines "Programming with C++", J.R.Hubbard, Second Edition, Tata McGrawHill
- 4) Object Oriented programming with C++ , E Balagurusamy , Third Edition ,Tata McGraw Hill.
- 5) Pure C++ programming , Amir Afzal, Pearson Education.
- 6) Computer Science – A structured Approach using C++ by B. Forouzan, R. F. Gilberg, Cengage Publication.

| | | |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| Course Code UIT1DEL | Digital Electronics | |
| <p>Objectives The objective of this course is to acquire the basic knowledge of digital logic levels and the application of knowledge to understand digital electronics circuits. To prepare the learners to perform the analysis and design of various digital electronic circuits.</p> <p>Expected Learning Outcomes</p> <ol style="list-style-type: none"> 1) Learners should be able to have a thorough understanding of the fundamental concepts and techniques used in digital electronics. 2) Learners should be able to understand and examine the structure of various number systems and its application in digital design. 3) Learners should be able to identify basic requirements for a design application and design various combinational and sequential circuits. | | |
| I | <p>Number System: Analog System, digital system, numbering system, binary number system, octal number system, hexadecimal number system, conversion from one number system to another, floating point numbers, weighted codes binary coded decimal, non-weighted codes Excess – 3 code, Gray code, Alphanumeric codes –ASCII Code, EBCDIC, ISCII Code, Hollerith Code, Morse Code, Teletypewriter (TTY), Error detection and correction, Universal Product Code, Code conversion.</p> <p>Binary Arithmetic: Binary addition, Binary subtraction, Negative number representation, Subtraction using 1’s complement and 2’s complement, Binary multiplication and division, Arithmetic in octal number system, Arithmetic in hexadecimal number system, BCD and Excess – 3 arithmetic.</p> | 12 |
| II | <p>Boolean Algebra and Logic Gates: Introduction, Logic (AND OR NOT), Boolean theorems, Boolean Laws, De Morgan’s Theorem, Perfect Induction, Reduction of Logic expression using Boolean Algebra, Deriving Boolean expression from given circuit, exclusive OR and Exclusive NOR gates, Universal Logic gates, Implementation of other gates using universal gates, Input bubbled logic, Assertion level.</p> <p>Minterm, Maxterm and Karnaugh Maps: Introduction, minterms and sum of minterm form, maxterm and Product of maxterm form, Reduction technique using Karnaugh maps – 2/3/4/5/6 variable K-maps, Grouping of variables in K-maps, K-maps for product of sum form, minimize Boolean expression using K-map and obtain K-map from Boolean expression, Quine Mc Cluskey Method</p> | 12 |
| III | <p>Combinational Logic Circuits: Introduction, Multi-input, multi-output Combinational circuits, Code converters design and implementations</p> <p>Arithmetic Circuits: Introduction, Adder, BCD Adder, Excess – 3 Adder, Binary Subtractors, BCD Subtractor, Multiplier, Comparator</p> <p>Introduction, Multi-input, multi-output Combinational circuits, Code converters design and implementations Arithmetic Circuits: Introduction, Adder, BCD Adder, Excess – 3 Adder, Binary Subtractors, BCD Subtractor, Multiplier, Comparator</p> | 12 |

| | | |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| IV | <p>Multiplexer, Demultiplexer, ALU, Encoder and Decoder: Introduction, Multiplexer, Demultiplexer, Decoder, ALU, Encoders.</p> <p>Sequential Circuits: Flip-Flop: Introduction, Terminologies used, S-R flip-flop, D flip-flop, JK flipflop, Race-around condition, Master – slave JK flip-flop, T flip-flop, 12 14 Page conversion from one type of flip-flop to another, Application of flipflops</p> | 12 |
| V | <p>Counters: Introduction, Asynchronous counter, Terms related to counters, IC 7493 (4-bit binary counter), Synchronous counter, Bushing, Type T Design, Type JK Design, Presetable counter, IC 7490, IC 7492, Synchronous counter ICs, Analysis of counter circuits.</p> <p>Shift Register: Introduction, parallel and shift registers, serial shifting, serial–in serial– out, serial–in parallel–out , parallel–in parallel–out, Ring counter, Johnson counter, Applications of shift registers, Pseudo-random binary sequence generator, IC7495, Seven Segment displays, analysis of shift counters.</p> | 12 |

| Course Code | Practical List |
|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| UIT1DEP | <p>1. Study of Logic gates and their ICs and universal gates:</p> <ol style="list-style-type: none"> Study of AND, OR, NOT, XOR, XNOR, NAND and NOR gates IC 7400, 7402, 7404, 7408, 7432, 7486, 74266 Implement AND, OR, NOT, XOR, XNOR using NAND gates. Implement AND, OR, NOT, XOR, XNOR using NOR gates. <p>2. Implement the given Boolean expressions using minimum number of gates.</p> <ol style="list-style-type: none"> Verifying De Morgan’s laws. Implement other given expressions using minimum number of gates. Implement other given expressions using minimum number of ICs. <p>3. Implement combinational circuits.</p> <ol style="list-style-type: none"> Design and implement combinational circuit based on the problem given and minimizing using K-maps. <p>4. Implement code converters.</p> <ol style="list-style-type: none"> Design and implement Binary – to – Gray code converter. Design and implement Gray – to – Binary code converter. Design and implement Binary – to – BCD code converter Design and implement Binary – to – XS-3 code converter <p>5. Implement Adder and Subtractor Arithmetic circuits.</p> <ol style="list-style-type: none"> Design and implement Half adder and Full adder. Design and implement BCD adder. Design and implement XS – 3 adder. Design and implement binary subtractor. Design and implement BCD subtractor. Design and implement XS – 3 subtractor. <p>6. Implement Arithmetic circuits.</p> <ol style="list-style-type: none"> Design and implement a 2-bit by 2-bit multiplier. Design and implement a 2-bit comparator. <p>7. Implement Encode and Decoder and Multiplexer and Demultiplexers.</p> <ol style="list-style-type: none"> Design and implement 8:3 encoder. Design and implement 3:8 decoder. Design and implement 4:1 multiplexer. Study of IC 74153, 74157 |

- d. Design and implement 1:4 demultiplexer. Study of IC 74139
- e. Implement the given expression using IC 74151 8:1 multiplexer.
- f. Implement the given expression using IC 74138 3:8 decoder.

8. Study of flip-flops and counters.

- a. Study of IC 7473.
- b. Study of IC 7474.
- c. Study of IC 7476.
- d. Conversion of Flip-flops.
- e. Design of 3-bit synchronous counter using 7473 and required gates.
- f. Design of 3-bit ripple counter using IC 7473.

9. Study of counter ICs and designing Mod-N counters.

- a. Study of IC 7490, 7492, 7493 and designing mod-n counters using these.
- b. Designing mod-n counters using IC 7473 and 7400 (NAND gates)

10. Design of shift registers and shift register counters.

- a. Design serial – in serial – out, serial – in parallel – out, parallel – in serial – out, parallel – in parallel – out and bidirectional shift registers using IC 7474.
- b. Study of ID 7495.
- c. Implementation of digits using seven segment displays.

Reference Books:

- 1) Digital Electronics and Logic Design, N. G. Palan, Technova
- 2) Make Electronics, Charles Platt, O'Reilly, 1st, 2010
- 3) Modern Digital Electronics, R. P. Jain, Tata McGraw Hill, 3rd
- 4) Digital Principles and Applications, Malvino and Leach, Tata McGraw Hill
- 5) Digital Electronics: Principles, Devices and Applications, Anil K. Maini, Wiley, 2007

| | | |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| Course Code UIT10PS | Operating Systems | |
| Objectives: | | |
| To learn the fundamentals of Operating Systems, its functions and services. To learn the mechanisms of OS to handle processes and threads and their communication .To learn the mechanisms involved in memory management in contemporary OS. | | |
| Expected Learning Outcomes: | | |
| 1) Learners should be able to analyze the structure of OS and basic architectural components involved in OS design. 2) Learners should be able to analyze and design the applications to run in parallel either using process or thread models of different OS. 3) Learners should be able to understand the Mutual exclusion, Deadlock detection and agreement protocols of Distributed operating system. | | |
| I | Introduction: What is an operating system? History of operating system, computer hardware, different operating systems, operating system concepts, system calls, operating system structure. Processes and Threads: Processes, threads, interprocess communication, scheduling, IPC problems. | 12 |
| II | Memory Management: No memory abstraction, memory abstraction: address spaces, virtual memory, page replacement algorithms, design issues for paging systems, implementation issues, and segmentation. | 12 |
| III | File Systems: Files, directories, file system implementation, file-system management and optimization, MS-DOS file system, UNIX V7 file system, CD ROM file system. | 12 |
| IV | Input-Output: Principles of I/O hardware, Principles of I/O software, I/O software layers, disks, clocks, user interfaces: keyboard, mouse, monitor, thin clients, power management Deadlocks: Resources, introduction to deadlocks, the ostrich algorithm, deadlock detection and recovery, deadlock avoidance, deadlock prevention, issues. | 12 |
| V | Virtualization and Cloud: History, requirements for virtualization, type 1 and 2 hypervisors, techniques for efficient virtualization, hypervisor microkernel, memory virtualization, I/O virtualization, Virtual appliances, virtual machines on multicore CPUs, Clouds. Multiple Processor Systems: Multiprocessors, multicomputer, distributed systems. | 12 |

| Course Code | Practical List |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| UIT1OSP | <p>1. Installation of virtual machine software.</p> <p>2. Installation of Linux operating system (RedHat / Ubuntu) on virtual machine.</p> <p>3. Installation of Windows operating system on virtual machine.</p> <p>4. Linux commands: Working with Directories:</p> <ul style="list-style-type: none"> a. pwd, cd, absolute and relative paths, ls, mkdir, rmdir, b. file, touch, rm, cp, mv, rename, head, tail, cat, tac, more, less, strings, chmod <p>5. Linux commands: Working with files:</p> <ul style="list-style-type: none"> a. ps, top, kill, pkill, bg, fg, b. grep, locate, find, locate. c. date, cal, uptime, w, whoami, finger, uname, man, df, du, free, whereis, which. d. Compression: tar, gzip. <p>6. Windows (DOS) Commands – 1</p> <ul style="list-style-type: none"> a. Date, time, prompt, md, cd, rd, path. b. Chkdsk, copy, xcopy, format, fidsk, cls, defrag, del, move. <p>7. Windows (DOS) Commands – 2</p> <ul style="list-style-type: none"> a. Diskcomp, diskcopy, diskpart, doskey, echo b. Edit, fc, find, rename, set, type, ver <p>8. Working with Windows Desktop and utilities</p> <ul style="list-style-type: none"> a. Notepad b. Wordpad c. Paint d. Taskbar e. Adjusting display resolution f. Using the browsers g. Configuring simple networking h. Creating users and shares <p>9. Working with Linux Desktop and utilities</p> <ul style="list-style-type: none"> a. The vi editor. b. Graphics c. Terminal d. Adjusting display resolution e. Using the browsers f. Configuring simple networking g. Creating users and shares <p>10. Installing utility software on Linux and Windows</p> |

Reference Books:

- 1) Modern Operating Systems, Andrew S. Tanenbaum and Herbert Bos, 4th Edition, Pearson Publishers
- 2) Operating System Concepts, Abraham Silberschatz and Peter B. Galvineg Gagne, 8th Edition, Wiley Publishers
- 3) Operating Systems – Internals and Design Principles, Willaim Stallings, 8th Edition, Pearson Publishers
- 4) Operating Systems, Godbole and Kahate, 3rd Edition, McGraw Hill Publishers

| | | |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| Course Code UIT1DMS | Discrete Mathematics | |
| <p>Objectives: The purpose of the course is to familiarize the prospective learners with mathematical structure that are fundamentally discrete. This course introduces set and functions, forming and solving recurrence relations and different counting principles. These concepts are useful to study or describe object or problems in computer algorithms and programming languages.</p> <p>Expected Learning Outcomes: 1) To provide overview of theory of discrete objects, starting with relations and partially ordered sets. 2) Study about recurrence relations, generating function and operation on them. 3) Give an understanding of graphs and trees which are widely use in software.</p> | | |
| I | <p>Set Theory: Definitions and the Element Method of Proof, Properties of Sets, Disproofs, Algebraic Proofs, Boolean Algebras, Russell’s Paradox and the Halting Problem. The Logic of Compound Statements: Logical Form and Logical Equivalence, Conditional Statements, Valid and Invalid Arguments. Quantified Statements: Predicates and Quantified Statements, Statements with Multiple Quantifiers, Arguments with Quantified Statements</p> | 12 |
| II | <p>Functions: Functions Defined on General Sets, One-to-One and Onto, Inverse Functions, Composition of Functions, Cardinality with Applications to Computability Relations: Relations on Sets, Reflexivity, Symmetry, and Transitivity, Equivalence Relations, Partial Order Relations Recursion Relation: solving recurrence relations by iteration, Second order linear homogenous recurrence relations with constant coefficients. general recursive definitions and structural induction.</p> | 12 |
| III | <p>Counting Principles and probability: Introduction, Possibility Trees and the Multiplication Rule, Counting Elements of Disjoint Sets: The Addition Rule, The Pigeonhole Principle, Counting Subsets of a Set: Combinations, r-Combinations with Repetition Allowed, Mathematical Induction, Strong Mathematical Induction and the Well-Ordering Principle for the Integers .Probability Axioms and Expected Value, Conditional Probability, Bayes’ Formula, and Independent Events.</p> | 12 |
| IV | <p>Graphs and Trees: Definitions and Basic Properties, Trails, Paths, and Circuits, Matrix Representations of Graphs, Isomorphism’s of Graphs, Trees, Rooted Trees, Isomorphism’s of Graphs, Spanning trees and shortest paths.</p> | 12 |
| V | <p>Elementary Number Theory and Methods of Proof: Introduction to Direct Proofs, Rational Numbers, Divisibility, Division into Cases and the Quotient-Remainder Theorem, Floor and Ceiling, Indirect Argument: Contradiction and Contraposition, Two Classical Theorems, Applications in algorithms.</p> | 12 |

| Course Code | Practical List | | | | | | |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| UIT1DMP | <p>Write the programs for the following using SCILAB</p> <ol style="list-style-type: none"> 1. Set Theory <ol style="list-style-type: none"> a. Inclusion Exclusion principle. b. Power sets c. Mathematical Induction 2. Functions and Algorithms <ol style="list-style-type: none"> a. Recursively defined functions b. Cardinality c. Polynomial evaluations d. Greatest Common Divisor 3. Recurrence Relation <ol style="list-style-type: none"> a. Linear homogeneous recurrence relations with constant coefficients b. Solving linear homogeneous recurrence relations with constant coefficients c. Solving general homogeneous linear recurrence relations 4. Counting <table border="0" style="width: 100%;"> <tr> <td style="vertical-align: top;"> <ol style="list-style-type: none"> a. Sum rule principle b. Product rule principle c. Factorial d. Binomial coefficients e. Permutations </td> <td style="vertical-align: top;"> <ol style="list-style-type: none"> f. Permutations with repetitions g. Combinations h. Combinations with repetitions i. Ordered partitions j. Unordered partitions </td> </tr> </table> 5. Probability Theory <table border="0" style="width: 100%;"> <tr> <td style="vertical-align: top;"> <ol style="list-style-type: none"> a. Sample space and events b. Finite probability spaces c. Equiprobable spaces d. Addition Principle </td> <td style="vertical-align: top;"> <ol style="list-style-type: none"> e. Conditional Probability f. Multiplication theorem for conditional probability g. Independent events h. Repeated trials with two outcomes </td> </tr> </table> 6. Graph Theory <ol style="list-style-type: none"> a. Paths and connectivity b. Minimum spanning tree c. Isomorphism 7. Direct Graphs <ol style="list-style-type: none"> a. Adjacency matrix b. Path matrix 8. Properties of integers <table border="0" style="width: 100%;"> <tr> <td style="vertical-align: top;"> <ol style="list-style-type: none"> a. Division algorithm b. Primes c. Euclidean algorithm </td> <td style="vertical-align: top;"> <ol style="list-style-type: none"> d. Fundamental theorem of arithmetic e. Congruence relation f. Linear congruence equation </td> </tr> </table> 9. Algebraic Systems <ol style="list-style-type: none"> a. Properties of operations 10. Boolean Algebra <ol style="list-style-type: none"> a. Basic definitions in Boolean Algebra b. Boolean algebra as lattices | <ol style="list-style-type: none"> a. Sum rule principle b. Product rule principle c. Factorial d. Binomial coefficients e. Permutations | <ol style="list-style-type: none"> f. Permutations with repetitions g. Combinations h. Combinations with repetitions i. Ordered partitions j. Unordered partitions | <ol style="list-style-type: none"> a. Sample space and events b. Finite probability spaces c. Equiprobable spaces d. Addition Principle | <ol style="list-style-type: none"> e. Conditional Probability f. Multiplication theorem for conditional probability g. Independent events h. Repeated trials with two outcomes | <ol style="list-style-type: none"> a. Division algorithm b. Primes c. Euclidean algorithm | <ol style="list-style-type: none"> d. Fundamental theorem of arithmetic e. Congruence relation f. Linear congruence equation |
| <ol style="list-style-type: none"> a. Sum rule principle b. Product rule principle c. Factorial d. Binomial coefficients e. Permutations | <ol style="list-style-type: none"> f. Permutations with repetitions g. Combinations h. Combinations with repetitions i. Ordered partitions j. Unordered partitions | | | | | | |
| <ol style="list-style-type: none"> a. Sample space and events b. Finite probability spaces c. Equiprobable spaces d. Addition Principle | <ol style="list-style-type: none"> e. Conditional Probability f. Multiplication theorem for conditional probability g. Independent events h. Repeated trials with two outcomes | | | | | | |
| <ol style="list-style-type: none"> a. Division algorithm b. Primes c. Euclidean algorithm | <ol style="list-style-type: none"> d. Fundamental theorem of arithmetic e. Congruence relation f. Linear congruence equation | | | | | | |

Reference Books:

- 1) Discrete Mathematics with Applications, Sussana S. Epp, 4th Edition, 2010
- 2) Discrete Mathematics, Schaum's Outlines Series, Seymour Lipschutz, Marc Lipson Tata McGraw Hill 2007
- 3) Discrete Mathematics and its Applications , Kenneth H. Rosen , Tata McGraw Hill
- 4) Discrete mathematical structures , B Kolman RC Busby, S Ross , PHI
- 5) Discrete structures , Liu , Tata McGraw Hill

| Course Code UIT1CMS | Communication Skills | |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| <p>Objectives: To understand the basics of Professional as well as Business Communication Skills.</p> <p>Expected Learning Outcomes:</p> <ol style="list-style-type: none"> 1) To Understand the basics of communication and to develop the communication skill. 2) To Understand the business communication. 3) To Make the learners aware about the communication skills at corporate level. | | |
| I | <p>Understanding Business Communication: Nature and Scope of Communication, Non-verbal Communication, Cross-cultural communication, Technology-enabled Business Communication</p> <p>The Seven Cs of Effective Communication: Completeness, Conciseness, Consideration, Concreteness, Clarity, Courtesy, Correctness</p> | 12 |
| II | <p>Writing Business Messages and Documents: Business writing, Business Correspondence, Instructions Business Reports and Proposals, Career building, Job Application and Resume writing.</p> | 12 |
| III | <p>Group Communication: Meetings and Conferences, Group Discussions and Team Presentations, Team Briefing , Understanding Specific Communication Needs, Communication across Functional Areas</p> | 12 |
| IV | <p>Understanding Specific Communication Needs: Corporate Communication, Persuasive Strategies in Business Communication, Ethics in Business Communication, Business Communication Aids</p> | 12 |
| V | <p>Presentation Skills: Planning the presentations, executing the presentations, Impressing the audience by performing, Planning stage: Brainstorming, mind maps / concept maps, executing stage: chunking theory, creating outlines, Use of templates. Adding graphics to your presentation: Visual communication, Impress stage: use of font, colour, layout, Importance of practice and performance.</p> | 12 |

| Course Code | Practical List |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | <p>1. Mock Interviews</p> <ol style="list-style-type: none"> a. You are in the HR department of an organization and you are supposed to hire a candidate. Write a telephonic interview between you and the client. b. Write an interview (a face-to-face interview) between Mr Smith and Mr John regarding a job position in a company. Mr John is the manager whereas Mr Smith is the candidate. c. You are asked to conduct a video interview for hiring a candidate in your company. Write the questions than you can ask and possible answers that can be given by the candidate. <p>2. Presentations</p> <ol style="list-style-type: none"> a. 3D Glasses b. Apple Talk c. Mobile IP d. Big Data |

3. Group Discussions

- a. Climate Change/ Pollution
- b. Banking Scams
- c. Poverty in India: Facts, Causes, Effects, Solution
- d. Social Media: Impact on human behaviour and society

4. Role Play

- a. Assume that you are a Sales person. Write a conversation between the sales person and the customer for selling a printer.
- b. Introduce yourself as a kitchen gadget
- c. Assume that you are a king of a kingdom. Write a conversation between you and your ministers regarding the development of your kingdom.

5. Situational Conversion

- a. Tell me about a time you proved you're the perfect person for this job.
- b. What would you do if you made a mistake that no one else noticed? Would you address the error and risk slowing things down or ignore it to keep the project or task moving forward?
- c. What would you do if you were asked to perform a task you've never done before?
- d. What would you do if an angry and dissatisfied customer confronted you? How would you resolve their concern?

6. Advertising

- a. There is a campaign in you college regarding the women's safety in college. Write an advertisement for the same.
- b. A new product named 'Techno' is introduced by an IT company which helps you locate your personal things like mobile phone, wallet, keys, etc. Write an advertisement to sell this product. Also state its features.
- c. There is new TV reality show and you are asked to promote it. How will you write an advertisement for the same.

7. Story-Telling

- a. A middle-aged woman discovers a ghost.
- b. A group of children discover a dead body
- c. A long journey is interrupted by a disaster.

8. Pronunciation Skills

9. Listening Skills

10. Writing Skills

Reference Books:

- 1) Business Communication, Edited by Meenakshi Raman and Prakash Singh, Second Edition, Oxford University Press,
- 2) Professional Communication, ArunaKoneru, Tata McGraw Hill
- 3) Business Communication, Dr.Rishipal and Dr.JyotiSheoran, SPD
- 4) Strategies for improving your business communication, Prof. M. S. Rao, Shroff publishers and distributors
- 5) Graphics for Learning: Proven Guidelines for Planning, Designing, and Evaluating Visuals in Training Materials, Ruth C. Clark, Chopeta Lyons, Pfeiffer

Semester- II

| | | |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| Course Code UIT2OOP | Object Oriented Programming | |
| <p>Objectives: To learn advanced features of the C++ programming language as a continuation of the previous course, to learn the characteristics of an object-oriented programming language: data abstraction and information hiding, inheritance, and dynamic binding of the messages to the methods.</p> <p>Expected Learning Outcomes:</p> <ol style="list-style-type: none"> 1) Use the characteristics of an object-oriented programming language in a program. 2) Use the basic object-oriented design principles in computer problem solving. 3) Use the basic principles of software engineering in managing complex software project. 4) Program with advanced features of the C++ programming language. | | |
| I | <p>Introduction to OOPs: Need object oriented programming, comparison of procedural and object oriented approach, characteristics of OOPs – object , classes , polymorphism, inheritance, reusability, data hiding and abstraction, applications of OOPs</p> <p>Classes and Objects: Class declaration, constructors, constructor initialization lists, access functions, private member functions, the copy constructor, the class destructor ,constant objects, structures, pointers to objects, static data members, static function members</p> | 12 |
| II | <p>Operator Overloading: overloading the assignment operator, the this pointer, overloading arithmetic operators, overloading the arithmetic assignment, operators, overloading the relational operators, overloading the stream operators, conversion operators ,overloading the increment and decrement operators, overloading the subscript operator</p> | 12 |
| III | <p>Composition and Inheritance: inheritance, protected class members, overriding and dominating inherited members, private access verses protected access, virtual functions and polymorphism, virtual destructors, abstract base classes</p> <p>File Handling: Classes for file stream operations, opening and closing a file, detecting end of file, file modes, file pointers and their manipulations, sequential input and output operations, random access, file operations error handling, command line argument</p> | 12 |
| IV | <p>Strings and Streams: the string class interface, the constructors and destructor , the copy constructor, the assignment operator, the addition operator , an append operator, access functions , the comparison operators, stream operators, stream classes, the ios class, ios format flags, ios state, variables, the istream and ostream classes, unformatted input functions, unformatted output functions, stream manipulators.</p> | 12 |
| V | <p>Templates and Iterators: function templates, class templates, container classes, subclass templates, passing template classes to template parameters, iterator classes</p> <p>Libraries: the standard C++ library, proprietary libraries, contents of the standard c headers, string streams, file processing, the standard template library</p> | 12 |

| Course Code | Practical List |
|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| UIT2OPP | <p>1. Classes and methods</p> <ol style="list-style-type: none"> Design an employee class for reading and displaying the employee information, the getInfo() and displayInfo() methods will be used respectively. Where getInfo() will be private method Design the class student containing getData() and displayData() as two of its methods which will be used for reading and displaying the student information respectively. Where getData() will be private method. Design the class Demo which will contain the following methods: readNo(), factorial() for calculating the factorial of a number, reverseNo() will reverse the given number, isPalindrome() will check the given number is palindrome, isArmstrong() which will calculate the given number is armStrong or not. Where readNo() will be private method. Write a program to demonstrate function definition outside class and accessing class members in function definition. <p>2. Using friend functions.</p> <ol style="list-style-type: none"> Write a friend function for adding the two complex numbers, using a single class Write a friend function for adding the two different distances and display its sum, using two classes. Write a friend function for adding the two matrix from two different classes and display its sum. <p>3. Constructors and method overloading.</p> <ol style="list-style-type: none"> Design a class Complex for adding the two complex numbers and also show the use of constructor. Design a class Geometry containing the methods area() and volume() and also overload the area() function . Design a class StaticDemo to show the implementation of static variable and static function. <p>4. Operator Overloading</p> <ol style="list-style-type: none"> Overload the operator unary(-) for demonstrating operator overloading. Overload the operator + for adding the timings of two clocks, And also pass objects as an argument. Overload the + for concatenating the two strings. For e.g “Py” + “thon” = Python <p>5. Inheritance</p> <ol style="list-style-type: none"> Design a class for single level inheritance using public and private type derivation. Design a class for multiple inheritances. Implement the hierarchical inheritance. <p>6. Virtual functions and abstract classes</p> <ol style="list-style-type: none"> Implement the concept of method overriding. Show the use of virtual function Show the implementation of abstract class. <p>7. String handling</p> <ol style="list-style-type: none"> String operations for string length , string concatenation String operations for string reverse, string comparison, Console formatting functions. <p>8. Exception handling</p> <ol style="list-style-type: none"> Show the implementation of exception handling Show the implementation for exception handling for strings <p>9. File handling</p> |

- | | |
|--|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | <ul style="list-style-type: none">a. Design a class FileDemo open a file in read mode and display the total number of words and lines in the file.b. Design a class to handle multiple files and file operationsc. Design an editor for appending and editing the files |
|--|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

10. Templates

- | | |
|--|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | <ul style="list-style-type: none">a. Design the template class library for concatenating two stringsb. Design the implementation of template class library for swap function.c. Design the template class library for sorting ascending to descending and vice-versa |
|--|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Reference Books:

- 1) Object Oriented Design by Rumbaugh (Pearson publication)
- 2) Object-oriented programming in Turbo C++ By Robert Lafore, Galgotia Publication.
- 3) Object-oriented programming with C++ by E.Balagurusamy, 2nd Edition, TMH.

| | | |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| Course Code USC2MPA | Microprocessor Architecture | |
| Objectives: To understand the basic structure of 8085 Microprocessor and its instruction set. | | |
| Expected Learning Outcomes: 1) To learn about how microprocessor works and its basic principles. 2) To understand the basics of assembly language and 8085 microprocessor . 3) To understand how data can be transferred between microprocessor and peripherals. | | |
| I | Microprocessor, microcomputers, and Assembly Language: Microprocessor, Microprocessor Instruction Set and Computer Languages, From Large Computers to Single-Chip Microcontrollers, Applications. Microprocessor Architecture and Microcomputer System: Microprocessor Architecture and its operation's, Memory, I/O Devices, Microcomputer System, Logic Devices and Interfacing, Microprocessor-Based System Application. Introduction to 8085 Assembly Language Programming: The 8085 Programming Model, Instruction Classification, Instruction, Data and Storage, Writing assembling and Execution of a simple program, Overview of 8085 Instruction Set, Writing and Assembling Program. | 12 |
| II | Introduction to 8085 Instructions: Data Transfer Operations, Arithmetic Operations, Logic Operation, Branch Operation, Writing Assembly Languages Programs, Debugging a Program 8085 Microprocessor Architecture and Memory Interface: Introduction, 8085 Microprocessor unit, 8085-Based Microcomputer, Memory Interfacing, Interfacing the 8155 Memory Segment, Illustrative Example: Designing Memory for the MCTS Project, Testing and Troubleshooting Memory Interfacing Circuit, 8085-Based Single-Board microcomputer. Interfacing of I/O Devices: Basic Interfacing concepts, Interfacing Output Displays, Interfacing Input Devices, Memory Mapped I/O, Testing and Troubleshooting I/O Interfacing Circuits. | 12 |
| III | Programming Techniques With Additional Instructions: Programming Techniques: Looping, Counting and Indexing, Additional Data Transfer and 16-Bit Arithmetic Instructions, Arithmetic Instruction Related to Memory, Logic Operations: Rotate, Logics Operations: Compare, Dynamic Debugging. Counters and Time Delays: Counters and Time Delays, Illustrative Program: Hexadecimal Counter, Illustrative Program: zero-to-nine (Modulo Ten) Counter, Generating Pulse Waveforms, Debugging Counter and Time-Delay Programs. | 12 |
| IV | Stacks and Sub-Routines: Stack Subroutine, Restart, Conditional Call, Return Instructions, Advanced Subroutine concepts. Code Conversion, BCD Arithmetic, and 16-Bit Data Operations: BCD-to-Binary Conversion, Binary-to-BCD Conversion, BCD-to-Seven-Segment-LED Code Conversion, Binary-to-ASCII and ASCII-to-Binary Code Conversion, BCD Addition, BCD Subtraction, Introduction To Advanced Instructions and Applications, Multiplication, Subtraction With Carry. | 12 |

| | | |
|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| V | <p>Software Development System and Assemblers: Microprocessors-Based Software Development system, Operating System and Programming Tools, Assemblers and Cross-Assemblers, Writing Program Using Cross Assemblers.</p> <p>Interrupts: The 8085 Interrupt, 8085 Vectored Interrupts, Restart as S/W Instructions, Additional I/O Concepts and processes.</p> | 12 |
|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|

| Course Code | Practical List |
|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| USC2MAP | <p>1. Perform the following Operations related to memory locations.</p> <ol style="list-style-type: none"> a. Store the data byte 32H into memory location 4000H. b. Exchange the contents of memory locations 2000H and 4000H <p>2. Simple assembly language programs.</p> <ol style="list-style-type: none"> a. Subtract the contents of memory location 4001H from the memory location 2000H and place the result in memory location 4002H. b. Subtract two 8-bit numbers. c. Add the 16-bit number in memory locations 4000H and 4001H to the 16-bit number in memory locations 4002H and 4003H. The most significant eight bits of the two numbers to be added are in memory locations 4001H and 4003H. Store the result in memory locations 4004H and 4005H with the most significant byte in memory location 4005H. d. Add the contents of memory locations 4000H and 4001H and place the result in the memory locations 4002H and 4003H. e. Subtract the 16-bit number in memory locations 4002H and 4003H from the 16-bit number in memory locations 4000H and 4001H. The most significant eight bits of the two numbers are in memory locations 4001H and 4003H. Store the result in memory locations 4004H and 4005H with the most significant byte in memory location 4005H. f. Find the 1's complement of the number stored at memory location 4400H and store the complemented number at memory location 4300H. g. Find the 2's complement of the number stored at memory location 4200H and store the complemented number at memory location 4300H. <p>3. Packing and unpacking operations.</p> <ol style="list-style-type: none"> a. Pack the two unpacked BCD numbers stored in memory locations 4200H and 4201H and store result in memory location 4300H. Assume the least significant digit is stored at 4200H. b. Two digit BCD number is stored in memory location 4200H. Unpack the BCD number and store the two digits in memory locations 4300H and 4301H such that memory location 4300H will have lower BCD digit. <p>4. Register Operations</p> <ol style="list-style-type: none"> a. Write a program to shift an eight bit data four bits right. Assume that data is in register C. b. Program to shift a 16-bit data 1 bit left. Assume data is in the HL register pair c. Write a set of instructions to alter the contents of flag register in 8085 d. Write a program to count number of 1's in the contents of D register and store the count in the B register. |

5. Multiple memory locations.

- a. Calculate the sum of series of numbers. The length of the series is in memory location 4200H and the series begins from memory location 4201H. a. Consider the sum to be 8 bit number. So, ignore carries. Store the sum at memory location 4300H. Consider the sum to be 16 bit number. Store the sum at memory locations 4300H and 4301H
- b. Multiply two 8-bit numbers stored in memory locations 2200H and 2201H by repetitive addition and store the result in memory locations 2300H and 2301H.
- c. Find the largest number in a block of data. The length of the block is in memory location 2200H and the block itself starts from memory location 2201H. Store the maximum number in memory location 2300H. Assume that the numbers in the block are all 8 bit unsigned binary numbers.

6. Calculations with respect to memory locations.

- a. Write a program to sort given 10 numbers from memory location 2200H in the ascending order.
- b. Calculate the sum of series of even numbers from the list of numbers. The length of the list is in memory location 2200H and the series itself begins from memory location 2201H. Assume the sum to be 8 bit number so you can ignore carries and store the sum at memory location 2300H. Sample problem:
- c. Calculate the sum of series of odd numbers from the list of numbers. The length of the list is in memory location 2200H and the series itself begins from memory location 2201H. Assume the sum to be 16-bit. Store the sum at memory locations 2300H and 2301H.
- d. Find the square of the given numbers from memory location 6100H and store the result from memory location 7000H
- e. Search the given byte in the list of 50 numbers stored in the consecutive memory locations and store the address of memory location in the memory locations 2200H and 2201H. Assume byte is in the C register and starting address of the list is 2000H. If byte is not found store 00 at 2200H and 2201H
- f. Add 2 arrays having ten 8-bit numbers each and generate a third array of result. It is necessary to add the first element of array 1 with the first element of array-2 and so on. The starting addresses of array 1, array2 and array3 are 2200H, 2300H and 2400H, respectively

7. Assembly programs on memory locations.

- a. Write an assembly language program to separate even numbers from the given list of 50 numbers and store them in the another list starting from 2300H. Assume starting address of 50 number list is 2200H
- b. Add even parity to a string of 7-bit ASCII characters. The length of the string is in memory location 2040H and the string itself begins in memory location 2041H. Place even parity in the most significant bit of each character.
- c. A list of 50 numbers is stored in memory, starting at 6000H. Find number of negative, zero and positive numbers from this list and store these results in memory locations 7000H, 7001H, and 7002H respectively
- d. Write an assembly language program to generate fibonacci number.

8. String operations in assembly programs.

- a. Write an 8085 assembly language program to insert a string of four characters from the tenth location in the given array of 50 characters
- b. Write an 8085 assembly language program to delete a string of 4 characters from the tenth location in the given array of 50 characters.
- c. Multiply the 8-bit unsigned number in memory location 2200H by the 8-bit

unsigned number in memory location 2201H. Store the 8 least significant bits of the result in memory location 2300H and the 8 most significant bits in memory location 2301H.

- d. DAA instruction is not present. Write a sub routine which will perform the same task as DAA.

9. Calculations on memory locations.

- a. To test RAM by writing '1' and reading it back and later writing '0' (zero) and reading it back. RAM addresses to be checked are 40FFH to 40FFH. In case of any error, it is indicated by writing 01H at port 10
- b. Arrange an array of 8 bit unsigned no in descending order
- c. Transfer ten bytes of data from one memory to another memory block. Source memory block starts from memory location 2200H where as destination memory block starts from memory location 2300H
- d. Write a program to find the Square Root of an 8 bit binary number. The binary number is stored in memory location 4200H and store the square root in 4201H

10. Operations on BCD numbers.

- a. Add two 4 digit BCD numbers in HL and DE register pairs and store result in memory locations, 2300H and 2301H. Ignore carry after 16 bit.
- b. Subtract the BCD number stored in E register from the number stored in the D register
- c. Write an assembly language program to multiply 2 BCD numbers

Reference Books:

- 1) Microprocessors Architecture, Programming and Applications with the 8085, Ramesh Gaonkar, Fifth Edition, PENRAM
- 2) Computer System Architecture, M. Morris Mano, PHI
- 3) Structured Computer Organization, Andrew C. Tanenbaum, PHI
- 4) "Computer Architecture: A Quantitative Approach" by J H Hennessy and D A Patterson
- 5) "The 8051 Microcontroller, Architecture, Programming and Applications" by Kenneth J Ayala
- 6) "Fundamentals of Microprocessors and Microcontrollers" by B Ram

| | | |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| CourseCode UIT2DMS | Database Management System | |
| Objectives The objective of this course is to introduce the concept of the DBMS with respect to the relational model, to specify the functional and data requirements for a typical database application and to understand creation, manipulation and querying of data in databases | | |
| Expected Learning Outcomes 1) Learners should be able to design the database schema with the use of appropriate data types for storage of data in database. 2) Learners should be able to create, manipulate, query and back up the databases. | | |
| I | Introduction: What is database system, purpose of database system, view of data, relational databases, database architecture, transaction management, Data Models: The importance of data models, Basic building blocks ,Business rules, The evolution of data models, Degrees of data abstraction. | 12 |
| II | Database Design, ER-Diagram and Unified Modeling Language: Database design and ER Model: overview, ER-Model, Constraints, ER Diagrams,ERD Issues, weak entity sets, Codd's rules, Relational Schemas, Introduction to UML Relational database model: Logical view of data, keys, integrity rules. Relational Database design: features of good relational database design,atomic domain and Normalization (1NF, 2NF, 3NF, BCNF). | 12 |
| II | Relational Algebra and Calculus: Relational algebra: introduction,Selection and projection, set operations, renaming, Joins, Division, syntax,semantics. Operators, grouping and ungrouping, relational comparison. Calculus: Tuple relational calculus, Domain relational Calculus, calculusvs algebra, computational capabilities. | 12 |
| IV | Constraints, Views and SQL: What is constraints, types of constrains,Integrity constraints, Views: Introduction to views, data independence,security, updates on views, comparison between tables and views SQL: data definition, aggregate function, Null Values, nested sub queries,Joined relations. Triggers. | 12 |
| V | Transaction management and Concurrency control: Transaction management: ACID properties, serializability and concurrency control,Lock based concurrency control (2PL, Deadlocks),Time stamping methods, optimistic methods, database recovery management. | 12 |

| Course Code | Practical List |
|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| UIT2DSP | 1. Design a Database and create required tables. a) Creating College database b) Creating Bank database 2. Writing Basic SQL SELECT Statements a) Restricting data b) Sorting Data 3. Applying the constraints a) Table Level b) Column Level 4. Single-Row Functions |

5. Manipulating Data

- a) Using INSERT
- b) Using UPDATE
- c) Using DELETE

6. Write a SQL statement for Creating and Managing Tables

- a) Alter
- b) Drop

7. Aggregating Data Using Group Functions

- a) SUM () , AVG ()
- b) MIN () , MAX ()
- c) COUNT

8. Write the queries to implement the joins

- a) Simple Join
- b) Outer Join

9. Write the queries to implement the set operators

- a) UNION , UNION ALL
- b) INTERSECT
- c) MINUS

10. Write the query to create the database objects

- a) Views
- b) Sequences

Reference Books:

- 1) "Database System and Concepts", A Silberschatz, H Korth, S Sudarshan, , fifth Edition McGraw-Hill ,
- 2) "Database Systems", Rob, Coronel, Seventh Edition, Cengage Learning

| | | |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| Course Code UIT2NMS | Numerical Methods | |
| Objectives: The course is designed to have a grasp of important concepts of Numerical Methods in a scientific way. The learner is expected to solve as many examples as possible to get complete clarity and understanding of the topics covered. | | |
| Expected Learning Outcomes: 1) Ability to appreciate real world applications which uses these concepts. 2) Skill to formulate a problem through Mathematical Modeling and simulation. | | |
| I | Solutions of Algebraic and Transcendental Equations: The Bisection Method, The Newton-Raphson Method, The Regula-falsi method, The Secant Method. Solution of simultaneous algebraic equation (linear) using iterative methods: Gauss Elimination Method, Gauss Jordan Method, Gauss Jacobi Method, Gauss Seidel Method | 12 |
| II | Interpolation: Forward Difference, Backward Difference, Central Differences, Different Types of Operators, Relation between Operators, Newton's Forward Difference Interpolation, Newton's Backward Difference Interpolation, Divided Differences, Newton's Divided Difference Interpolation, Lagrange's Interpolation, Spline Interpolation. | 12 |
| III | Numerical differentiation: Numerical differentiation, Methods based on finite Differences: Derivatives using Newton's Forward Difference Interpolation, Newton's Backward Difference Interpolation, Newton's Divided Difference Interpolation & Lagrange's Interpolation Numerical solution of 1st and 2nd order differential equations: Taylor series, Picard's Method, Euler's Method, Modified Euler's Method, Runge-Kutta Method for 1st and 2nd Order Differential Equations. | 12 |
| IV | Numerical integration: Trapezoidal Rule, Simpson's 1/3rd and 3/8th rules, Romberg Method, Gauss Legendre Integration Method, Gauss Chebyshev Integration Method Double Integration: Trapezoidal Method, Simpson's Method | 12 |
| V | Linear System of equations & Eigen Value Problems: Direct Method: Triangularization Method, LU Decomposition, Cholesky Method, Partition Method Eigen value Problem : Power Method | 12 |

| | |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Course Code | Practical List |
| UIT2NMP | 1. Solution of algebraic and transcendental equations: a. Program to solve algebraic and transcendental equation by bisection method. b. Program to solve algebraic and transcendental equation by false position method. c. Program to solve algebraic and transcendental equation by Secant method. d. Program to solve algebraic and transcendental equation by Newton |

| | |
|--|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | <p>Raphson method</p> <p>2. Solving linear system of equations by iterative methods</p> <p>a. Program for solving linear system of equations using Gauss Jordan method.</p> <p>b. Program for solving linear system of equations using Gauss Seidel method.</p> <p>3. Interpolation I</p> <p>a. Program for Newton's forward interpolation.</p> <p>b. Program for Newton's backward interpolation.</p> <p>c. Program for Newton's Divided Interpolation.</p> <p>4. Interpolation II:</p> <p>a. Program for Lagrange's interpolation.</p> <p>b. Program for Spline interpolation.</p> <p>5. Numerical Differentiation</p> <p>a. Programing to obtain derivatives numerically.</p> <p>6. Solution of differential equations</p> <p>a. Program to solve differential equation using Euler's method</p> <p>b. Program to solve differential equation using modified Euler's method.</p> <p>c. Program to solve differential equation using Runge-kutta 2nd order and 4th order methods</p> <p>7. Numerical Integration</p> <p>a. Program for numerical integration using Trapezoidal rule.</p> <p>b. Program for numerical integration using Simpson's 1/3rd rule.</p> <p>c. Program for numerical integration using Simpson's 3/8th rule.</p> <p>8. Double Integration:</p> <p>a. Program for numerical integration using Trapezoidal rule.</p> <p>b. Program for numerical integration using Simpson's rule.</p> <p>9. Linear System of Equation:</p> <p>a. Program for LU Decomposition</p> <p>b. Program for Partition Method</p> <p>10. Eigen Value Problem:</p> <p>a. Program for Power Method</p> |
|--|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Reference Books:

- 1) Numerical Methods for Scientific and Engineering Computation, M. K. Jain, S. R. K. Iyengar and R. K. Jain, New age International Publishers, Fourth Edition, 2003
- 2) Introductory Methods of Numerical Methods , S. S. Shastri , PHI , Vol – 2
- 3) Numerical Methods for Engineers , Steven C. Chapra, Raymond P. Canale , Tata Mc Graw Hill , 6th Edition , 2010
- 4) Numerical Analysis , Richard L. Burden, J. Douglas Faires , Cengage Learning , 9th Edition, 2011
- 5) Numerical and Statistical Technique, QaziShoeb Ahmad, Zubair Khan, Shadad Ahmad Khan, Ane's Student Edition

| | | |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| Course Code UIT2WPR | Web Programming | |
| <p>Objectives: On completion of this course, a learner will be able to develop a web application using web technologies. Learners will gain the skills and project-based experience needed for entry into web application and development careers. Learners will be able to develop a dynamic webpage by the use of java script , basic php along with interaction with mysql database.</p> <p>Expected Learning Outcomes: 1) Learners will be able to develop static web pages using HTML. 2) Learners will be able to add interactivity to web pages using javascript event handling and functions. 3) Learners will be able to retrieve form data sent from client, process it and store it on database using php code.</p> | | |
| I | <p>Internet and the World Wide Web: What is Internet? Introduction to internet and its applications, E-mail, telnet, FTP, e-commerce, video conferencing, e-business. Internet service providers, domain name server, internet address, World Wide Web (WWW): World Wide Web and its evolution, uniform resource locator (URL), browsers – internet explorer, Netscape navigator, opera, Firefox, chrome, Mozilla. search engine, web saver – apache, IIS, proxy server, HTTP protocol HTML5: Introduction, Why HTML5? Formatting text by using tags, using lists and backgrounds, Creating hyperlinks and anchors. Style sheets, CSS formatting text using style sheets, formatting paragraphs using style sheets</p> | 12 |
| II | <p>HTML5 Page layout and navigation: Creating navigational aids: planning site organization, creating text based navigation bar, creating graphics based navigation bar, creating graphical navigation bar, creating image map, redirecting to another URL, creating division based layouts: HTML5 semantic tags, creating divisions, creating HTML5 semantic layout, positioning and formatting divisions. HTML5 Tables, Forms and Media: Creating tables: creating simple table, specifying the size of the table, specifying the width of the column, merging table cells, using tables for page layout, formatting tables: applying table borders, applying background and foreground fills, changing cell padding, spacing and alignment, creating user forms: creating basic form, using check boxes and option buttons, creating lists, additional input types in HTML5, Incorporating sound and video: audio and video in HTML5, HTML multimedia basics, embedding video clips, incorporating audio on web page.</p> | 12 |
| III | <p>Java Script: Introduction, Client-Side JavaScript, Server-Side JavaScript, JavaScript Objects, JavaScript Security, Operators: Assignment Operators, Comparison Operators, Arithmetic Operators, % (Modulus), ++(Increment), --(Decrement), -(Unary Negation), Logical Operators, Short-Circuit Evaluation, String Operators, Special Operators, ?: (Conditional operator), , (Comma operator), delete, new, this, void Statements: Break, comment, continue, delete, do...while, export, for, for...in, function, if...else, import, labelled, return, switch, var, while, with Core JavaScript (Properties and Methods of Each):</p> | 12 |

| | | |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| | <p>Array, Boolean, Date, Function, Math, Number, Object, String, RegExp</p> <p>Document and its associated objects: Document, Link, Area, Anchor, Image, Applet, Layer</p> <p>Events and Event Handlers: General Information about Events, Defining Event Handlers, event, onAbort, onBlur, onChange, onClick, onDbClick, onDragDrop, onError, onFocus, onKeyDown, onKeyPress, onKeyUp, onLoad, onMouseDown, onMouseMove, onMouseOut, onMouseOver, onMouseUp, onMove, onReset, onResize, onSelect, onSubmit, onUnload</p> | |
| IV | <p>PHP: Why PHP and MySQL? Server-side scripting, PHP syntax and variables, comments, types, control structures, branching, looping, termination, functions, passing information with PHP, GET, POST, formatting form variables, superglobal arrays, strings and string functions, regular expressions, arrays, number handling, basic PHP errors/problems</p> | 12 |
| V | <p>Advanced PHP and MySQL: PHP/MySQL Functions, Integrating web forms and databases, Displaying queries in tables, Building Forms from queries, String and Regular Expressions, Sessions, Cookies and HTTP, E-Mail</p> | 12 |

| Course Code | Practical List |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| UIT2WRP | <p>1. Use of Basic Tags</p> <ol style="list-style-type: none"> Design a web page using different text formatting tags. Design a web page with links to different pages and allow navigation between web pages. Design a web page demonstrating all Style sheet types <p>2. Image maps, Tables, Forms and Media</p> <ol style="list-style-type: none"> Design a web page with Imagemaps. Design a web page demonstrating different semantics Design a web page with different tables. Design a webpages using table so that the content appears well placed. Design a web page with a form that uses all types of controls. Design a web page embedding with multimedia features. <p>3. Java Script</p> <ol style="list-style-type: none"> Using JavaScript design, a web page that prints factorial/Fibonacci series/any given series. Design a form and validate all the controls placed on the form using Java Script. Write a JavaScript program to display all the prime numbers between 1 and 100. Write a JavaScript program to accept a number from the user and display the sum of its digits. Write a program in JavaScript to accept a sentence from the user and display the number of words in it. (Do not use split () function). Write a java script program to design simple calculator. <p>4. Control and looping statements and Java Script references</p> <ol style="list-style-type: none"> Design a web page demonstrating different conditional statements. Design a web page demonstrating different looping statements. |

- c. Design a web page demonstrating different Core JavaScript references (Array, Boolean, Date, Function, Math, Number, Object, String, regExp).

5. Basic PHP I

- a. Write a PHP Program to accept a number from the user and print it factorial.
b. Write a PHP program to accept a number from the user and print whether it is prime or not.

6. Basic PHP II

- a. Write a PHP code to find the greater of 2 numbers. Accept the no. from the user.
b. Write a PHP program to display the following Binary Pyramid:

```
1
0 1
1 0 1
0 1 0 1
```

7.String Functions and arrays

- a. Write a PHP program to demonstrate different string functions.
b. Write a PHP program to create one dimensional array.

8.PHP and Database

- a. Write a PHP code to create:
• Create a database College
• Create a table Department (Dname, Dno, Number_Of_faculty)
b. Write a PHP program to create a database named “College”. Create a table named “Student” with following fields (sno, sname, percentage). Insert 3 records of your choice. Display the names of the students whose percentage is between 35 to 75 in a tabular format.
c. Design a PHP page for authenticating a user.

9.Email

- a. Write a program to send email with attachment.

10.Sessions and Cookies

- a. Write a program to demonstrate use of sessions and cookies.

Reference Books:

- 1) “HTML5 Step by Step”, FaitheWempen , Microsoft Press
- 2) “JavaScript 2.0: The Complete Reference “,Thomas Powell and Fritz Schneider , Second Edition, Tata McGraw Hill Publication
- 3) “PHP 5.1 for Beginners” , Ivan Bayross , Sharanam Shah, SPD Publication
- 4) “PHP 6 and MySQL Bible “,SteveSuehring, Tim Converse, Joyce Park, Wiley Publication
- 5) “PHP Project for Beginners “,SharanamShah, Vaishali Shah , SPD Publication
- 6) “Web Design The Complete Reference ” , Thomas Powell , Tata McGraw Hill Publication
- 7) “Head First HTML 5 programming “,Eric Freeman , O’Reilly Publication