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	(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. (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. (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. OR 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.
3	Passing marks	40%

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
UIT1CPT	Core Subject	Introduction to C++ programming	2
UIT1DET	Core Subject	Digital Electronics	2
UIT1OST	Core Subject	Operating Systems	2
UIT1DMT	Core Subject	Discrete Mathematics	2
UIT1CST	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
UIT1PCP	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
UIT2OPT	Core Subject	Object Oriented Programming	2
UIT2MAT	Core Subject	Microprocessor Architecture	2
UIT2DST	Core Subject	Database Management System	2
UIT2NMT	Core Subject	Numerical Methods	2
UIT2WPT	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
UIT2WPP	Ability Enhancement Skill Course Practical	Web Programming Practical	2
	Total Credits		**20**

Semester I

Course Code UIT1CPT	**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	**1. Basic Programs:** a. Write a program to display the message HELLO WORLD. b. Write a program to declare some variables of type int, float and double. Assign some values to these variables & display these values. c. Write a program to find the addition, subtraction, multiplication and division of two numbers. **2. Programs on variables:** a. Write a program to swap two numbers without using third variable. b. Write a program to find the area of rectangle, square and circle. c. Write a program to find the volume of a cube, sphere, and cylinder. **3. Conditional statements and loops(basic)** 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.

	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 UIT1DET	**Digital Electronics**	

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.

Expected Learning Outcomes

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	**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. **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.	**12**
II	**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. **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	**12**
III	**Combinational Logic Circuits:** 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, ComparatorIntroduction, 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	**12**

IV	**Multiplexer, Demultiplexer, ALU, Encoder and Decoder:** Introduction, Multiplexer, Demultiplexer, Decoder, ALU, Encoders. **Sequential Circuits:** Flip-Flop: Introduction, Terminologies used, S-R flip-flop, D flip-fop, 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	**12**
V	**Counters:** Introduction, Asynchronous counter, Terms related to counters, IC 7493 (4-bit binary counter), Synchronous counter, Bushing, Type T Design, Type JK Design, Presettable counter, IC 7490, IC 7492, Synchronous counter ICs, Analysis of counter circuits. **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.	**12**

Course Code	**Practical List**
UIT1DEP	**1. Study of Logic gates and their ICs and universal gates:** a. Study of AND, OR, NOT, XOR, XNOR, NAND and NOR gates b. IC 7400, 7402, 7404, 7408, 7432, 7486, 74266 c. Implement AND, OR, NOT, XOR, XNOR using NAND gates. d. Implement AND, OR, NOT, XOR, XNOR using NOR gates. **2. Implement the given Boolean expressions using minimum number of gates.** a. Verifying De Morgan's laws. b. Implement other given expressions using minimum number of gates. c. Implement other given expressions using minimum number of ICs. **3. Implement combinational circuits.** a. Design and implement combinational circuit based on the problem given and minimizing using K-maps. **4. Implement code converters.** a. Design and implement Binary – to – Gray code converter. b. Design and implement Gray – to – Binary code converter. c. Design and implement Binary – to – BCD code converter d. Design and implement Binary – to – XS-3 code converter **5. Implement Adder and Subtractor Arithmetic circuits.** a. Design and implement Half adder and Full adder. b. Design and implement BCD adder. c. Design and implement XS – 3 adder. d. Design and implement binary subtractor. e. Design and implement BCD subtractor. f. Design and implement XS – 3 subtractor. **6. Implement Arithmetic circuits.** a. Design and implement a 2-bit by 2-bit multiplier. b. Design and implement a 2-bit comparator. **7. Implement Encode and Decoder and Multiplexer and Demultiplexers.** a. Design and implement 8:3 encoder. b. Design and implement 3:8 decoder. c. 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 UIT1OST	**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	**1. Installation of virtual machine software.** **2. Installation of Linux operating system (RedHat / Ubuntu) on virtual machine.** **3. Installation of Windows operating system on virtial machine.** **4. Linux commands: Working with Directories:** 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 **5. Linux commands: Working with files:** 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. **6. Windows (DOS) Commands – 1** a. Date, time, prompt, md, cd, rd, path. b. Chkdsk, copy, xcopy, format, fidsk, cls, defrag, del, move. **7. Windows (DOS) Commands – 2** a. Diskcomp, diskcopy, diskpart, doskey, echo b. Edit, fc, find, rename, set, type, ver **8. Working with Windows Desktop and utilities** 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 **9. Working with Linux Desktop and utilities** 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 **10. Installing utility software on Linux and Windows**

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 UIT1DMT	**Discrete Mathematics**

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.

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.

I	**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	**12**
II	**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.	**12**
III	**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.	**12**
IV	**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.	**12**
V	**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.	**12**

Course Code	Practical List
UIT1DMP	**Write the programs for the following using SCILAB** **1. Set Theory** a. Inclusion Exclusion principle. b. Power sets c. Mathematical Induction **2. Functions and Algorithms** a. Recursively defined functions b. Cardinality c. Polynomial evaluations d. Greatest Common Divisor **3. Recurrence Relation** 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** a. Sum rule principle b. Product rule principle c. Factorial d. Binomial coefficients e. Permutations f. Permutations with repetitions g. Combinations h. Combinations with repetitions i. Ordered partitions j. Unordered partitions **5. Probability Theory** a. Sample space and events b. Finite probability spaces c. Equiprobable spaces d. Addition Principle e. Conditional Probability f. Multiplication theorem for conditional probability g. Independent events h. Repeated trials with two outcomes **6. Graph Theory** a. Paths and connectivity b. Minimum spanning tree c. Isomorphism **7. Direct Graphs** a. Adjacency matrix b. Path matrix **8. Properties of integers** a. Division algorithm b. Primes c. Euclidean algorithm d. Fundamental theorem of arithmetic e. Congruence relation f. Linear congruence equation **9. Algebraic Systems** a. Properties of operations **10. Boolean Algebra** a. Basic definitions in Boolean Algebra b. Boolean algebra as lattices

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 UIT1CST	**Communication Skills**

Objectives:
To understand the basics of Professional as well as Business Communication Skills.

Expected Learning Outcomes:
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	**Understanding Business Communication:** Nature and Scope of Communication, Non-verbal Communication, Cross-cultural communication, Technology-enabled Business Communication **The Seven Cs of Effective** Communication: Completeness, Conciseness, Consideration, Concreteness, Clarity, Courtesy, Correctness	**12**
II	**Writing Business Messages and Documents:** Business writing, Business Correspondence, Instructions Business Reports and Proposals, Career building, Job Application and Resume writing.	**12**
III	**Group Communication:** Meetings and Conferences, Group Discussions and Team Presentations, Team Briefing , Understanding Specific Communication Needs, Communication across Functional Areas	**12**
IV	**Understanding Specific Communication Needs:** Corporate Communication, Persuasive Strategies in Business Communication, Ethics in Business Communication, Business Communication Aids	**12**
V	**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.	**12**

Course Code	**Practical List**
UIT1PCP	**1. Mock Interviews** 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. **2. Presentations** 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 UIT2OPT	**Object Oriented Programming**

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.

Expected Learning Outcomes:
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	**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 **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	**12**
II	**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	**12**
III	**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 **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	**12**
IV	**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.	**12**
V	**Templates and Iterators:** function templates, class templates, container classes, subclass templates, passing template classes to template parameters, iterator classes **Libraries:** the standard C++ library, proprietary libraries, contents of the standard c headers, string streams, file processing, the standard template library	**12**

Course Code	Practical List
UIT2OPP	**1. Classes and methods** a. Design an employee class for reading and displaying the employee information, the getInfo() and displayInfo() methods will be used repectively. Where getInfo() will be private method b. 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. c. 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.WhereareadNo() will be private method. d. Write a program to demonstrate function definition outside class and accessing class members in function definition. **2. Using friend functions.** a. Write a friend function for adding the two complex numbers, using a single class b. Write a friend function for adding the two different distances and display its sum, using two classes. c. Write a friend function for adding the two matrix from two different classes and display its sum. **3. Constructors and method overloading.** a. Design a class Complex for adding the two complex numbers and also show the use of constructor. b. Design a class Geometry containing the methods area() and volume() and also overload the area() function . c. Design a class StaticDemo to show the implementation of static variable and static function. **4. Operator Overloading** a. Overload the operator unary(-) for demonstrating operator overloading. b. Overload the operator + for adding the timings of two clocks, And also pass objects as an argument. c. Overload the + for concatenating the two strings. For e.g “Py” + “thon” = Python **5. Inheritance** a. Design a class for single level inheritance using public and private type derivation. b. Design a class for multiple inheritances. c. Implement the hierarchical inheritance. **6. Virtual functions and abstract classes** a. Implement the concept of method overriding. b. Show the use of virtual function c. Show the implementation of abstract class. **7. String handling** a. String operations for string length , string concatenation b. String operations for string reverse, string comparison, c. Console formatting functions. **8. Exception handling** a. Show the implementation of exception handling b. Show the implementation for exception handling for strings **9. File handling**

	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 operations c. Design a editor for appending and editing the files **10. Templates** a. Design the template class library for concatenating two strings b. 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 UIT2MAT	**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	**Software Development System and Assemblers:** Microprocessors-Based Software Development system, Operating System and Programming Tools, Assemblers and Cross-Assemblers, Writing Program Using Cross Assemblers. **Interrupts:** The 8085 Interrupt, 8085 Vectored Interrupts, Restart as S/W Instructions, Additional I/O Concepts and processes.	**12**

Course Code	**Practical List**
USC2MAP	**1. Perform the following Operations related to memory locations.** a. Store the data byte 32H into memory location 4000H. b. Exchange the contents of memory locations 2000H and 4000H **2. Simple assembly language programs.** 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 40001H and 4001H and place the result in the memory locations 4002Hand 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 l'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. **3. Packing and unpacking operations.** 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. **4. Register Operations** 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 l'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 2Sample 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 l, 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 UIT2DST	**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 UMLRelational 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 UIT2NMT	**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 a 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

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

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 UIT2WPT	**Web Programming**	
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. **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.		
I	**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	**12**
II	**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.	**12**
III	**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)**:	**12**

	Array, Boolean, Date, Function, Math, Number, Object, String, regExp **Document and its associated objects:** Document, Link, Area, Anchor, Image, Applet, Layer **Events and Event Handlers:** General Information about Events, Defining Event Handlers, event, onAbort, onBlur, onChange, onClick, onDblClick, onDragDrop, onError, onFocus, onKeyDown, onKeyPress, onKeyUp, onLoad, onMouseDown, onMouseMove, onMouseOut, onMouseOver, onMouseUp, onMove, onReset, onResize, onSelect, onSubmit, onUnload	
IV	**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	**12**
V	**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	**12**

Course Code	**Practical List**
UIT2WPP	**1. Use of Basic Tags** a. Design a web page using different text formatting tags. b. Design a web page with links to different pages and allow navigation between web pages. c. Design a web page demonstrating all Style sheet types **2. Image maps, Tables, Forms and Media** a. Design a web page with Imagemaps. b. Design a web page demonstrating different semantics c. Design a web page with different tables. Design a webpages using table so that the content appears well placed. d. Design a web page with a form that uses all types of controls. e. Design a web page embedding with multimedia features. **3. Java Script** a. Using JavaScript design, a web page that prints factorial/Fibonacci series/any given series. b. Design a form and validate all the controls placed on the form using Java Script. c. Write a JavaScript program to display all the prime numbers between 1 and 100. d. Write a JavaScript program to accept a number from the user and display the sum of its digits. e. 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). f. Write a java script program to design simple calculator. **4. Control and looping statements and Java Script references** a. Design a web page demonstrating different conditional statements. b. 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

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 S.Y.B.Sc. Information Technology
Choice Based Credit & Grading System (60:40)
w.e.f. Academic Year 2020-2021

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 analyse 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)
S.Y.B. Sc. Information Technology Syllabus
To be implemented from the Academic year 2020-2021
SEMESTER III

Course Code	Course Type	Course Title	Credits
UIT3PYP	Skill Enhancement Course	Python Programming	2
UIT3DST	Core Subject	Data Structures	2
UIT3CNT	Core Subject	Computer Networks	2
UIT3DMS	Core Subject	Advanced SQL	2
UIT3MAT	Core Subject	Applied Mathematics	2
UIT3PPP	Skill Enhancement Course Practical	Python Programming Practical	2
UIT3DSP	Core Subject Practical	Data Structures Practical	2
UIT3CNP	Core Subject Practical	Computer Networks Practical	2
UIT3 DMP	Core Subject Practical	Advanced SQL Practical	2
UIT3MPP	Core Subject Practical	Mobile Programming Practical	2
	Total Credits		**20**

Choice Based Credit Grading and Semester System (CBCGS)
S.Y.B. Sc. Information Technology Syllabus
To be implemented from the Academic year 2020-2021
SEMESTER IV

Course Code	Course Type	Course Title	Credits
UIT4CJT	Skill Enhancement Course	Core Java	2
UIT4EMB	Core Subject	Introduction to Embedded systems	2
UIT4COS	Core Subject	Computer Oriented Statistical Techniques	2
UIT4SWE	Core Subject	Software Engineering	2
UIT4CGA	Core Subject	Computer Graphics and Animation	2
UIT4CJP	Skill Enhancement Course Practical	Core Java Practical	2
UIT4ESP	Core Subject Practical	Introduction to Embedded Systems Practical	2
UIT4COP	Core Subject Practical	Computer Oriented Statistical Techniques Practical	2
UIT4SEP	Core Subject Practical	Software Engineering Practical	2
UIT4CGP	Core Subject Practical	Computer Graphics and Animation Practical	2
	Total Credits		**20**

Semester III

Course Code UIT3PYP	**Python 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	**Introduction:** The Python Programming Language, History, features, Installing Python, Running Python program, Debugging : Syntax Errors, Runtime Errors, Semantic Errors, Experimental Debugging, Formal and Natural Languages, The Difference Between Brackets, Braces, and Parentheses, **Variables and Expressions** Values and Types, Variables, Variable Names and Keywords, Type conversion, Operators and Operands, Expressions, Interactive Mode and Script Mode, Order of Operations. **Conditional Statements:** if, if-else, nested if –else **Looping**: for, while, nested loops **Control statements:** Terminating loops, skipping specific conditions	**12**
II	**Functions:** Function Calls, Type Conversion Functions, Math Functions, Composition, Adding New Functions, Definitions and Uses, Flow of Execution, Parameters and Arguments, Variables and Parameters Are Local, Stack Diagrams, Fruitful Functions and Void Functions, Why Functions? Importing with from, Return Values, Incremental Development, Composition, Boolean Functions, More Recursion, Leap of Faith, Checking Types **Strings:** A String Is a Sequence, Traversal with a for Loop, String Slices, Strings Are Immutable, Searching, Looping and Counting, String Methods, The in Operator, String Comparison, String Operations.	**12**
III	**Lists:** Values and Accessing Elements, Lists are mutable, traversing a List, Deleting elements from List, Built-in List Operators, Concatenation, Repetition, In Operator, Built-in List functions and methods **Tuples and Dictionaries:** Tuples, Accessing values in Tuples, Tuple Assignment, Tuples as return values, Variable-length argument tuples, Basic tuples operations, Concatenation, Repetition, in Operator, Iteration, Built-in Tuple Functions Creating a Dictionary, Accessing Values in a dictionary, Updating Dictionary, Deleting Elements from Dictionary, Properties of Dictionary keys, Operations in Dictionary, Built-In Dictionary Functions, Built-in Dictionary Methods **Files:** Text Files, The File Object Attributes, Directories **Exceptions:** Built-in Exceptions, Handling Exceptions, Exception with Arguments, User-defined Exceptions	**12**
IV	**Regular Expressions –** Concept of regular expression, various types of regular expressions, using match function. **Classes and Objects:** Overview of OOP (Object Oriented Programming), Class Definition, Creating Objects, Instances as Arguments, Instances as return values, Built-in Class Attributes, Inheritance, Method Overriding, Data Encapsulation, Data Hiding **Multithreaded Programming:** Thread Module, creating a thread, synchronizing threads,	**12**

	multithreaded priority queue **Modules:** Importing module, Creating and exploring modules, Math module, Random module, Time module	
V	**Creating the GUI Form and Adding Widgets:** **Widgets:** Button, Canvas, Checkbutton, Entry, Frame, Label, Listbox, Menubutton, Menu, Message, Radiobutton, Scale, Scrollbar, text, Toplevel, Spinbox, PanedWindow, LabelFrame, tkMessagebox. Handling Standard attributes and Properties of Widgets. **Layout Management:** Designing GUI applications with proper Layout Management features. **Look and Feel Customization**: Enhancing Look and Feel of GUI using different appearances of widgets. **Storing Data in Our MySQL Database via Our GUI:** Connecting to a MySQL database from Python, Configuring the MySQL connection, Designing the Python GUI database, Using the INSERT command, Using the UPDATE command, Using the DELETE command, Storing and retrieving data from MySQL database.	**12**

Course Code	**Practical List**
UIT3PPP	**1. Write the program for the following:** a. Create a program that asks the user to enter their name and their age. Print out a message addressed to them that tells them the year that they will turn 100 years old. b. Enter the number from the user and depending on whether the number is even or odd, print out an appropriate message to the user. c. Write a program to generate the Fibonacci series. d. Write a function that reverses the user defined value. e. Write a function to check the input value is Armstrong and also write the function for Palindrome. f. Write a recursive function to print the factorial for a given number. **2. Write the program for the following:** a. Write a function that takes a character (i.e. a string of length 1) and returns True if it is a vowel, False otherwise. b. Define a function that computes the *length* of a given list or string. c. Define a procedurehistogram() that takes a list of integers and prints a histogram to the screen. For example, histogram([4, 9, 7]) should print the following: **** ********* ******* **3. Write the program for the following:** a. A pangram is a sentence that contains all the letters of the English alphabet at least once, for example: The quick brown fox jumps over the lazy dog. Your task here is to write a function to check a sentence to see if it is a pangram or not. b. Take a list, say for example this one: a=[1,1,2,3,5,8,13,21,34,55,89] and write a program that prints out all the elements of the list that are less than 5.

4. Write the program for the following:

a. Write a program that takes two lists and returns True if they have at least one common member.
b. Write a Python program to print a specified list after removing the 0th, 2nd, 4th and 5th elements.
c. Write a Python program to clone or copy a list

5. Write the program for the following:

a. Write a Python script to sort (ascending and descending) a dictionary by value.

b. Write a Python script to concatenate following dictionaries to create a new one.
Sample Dictionary : dic1={1:10, 2:20} dic2={3:30, 4:40} dic3={5:50,6:60}
Expected Result : {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}
c. Write a Python program to sum all the items in a dictionary.

6. Write the program for the following:

a. Write a Python program to read an entire text file.
b. Write a Python program to append text to a file and display the text.
c. Write a Python program to read last n lines of a file.

7. Write the program for the following:

a. Design a class that store the information of student and display the same
b. Implement the concept of inheritance using python

8. Write the program for the following:

a. Open a new file in IDLE ("New Window" in the "File" menu) and save it as geometry.py in the directory where you keep the files you create for this course. Then copy the functions you wrote for calculating volumes and areas in the "Control Flow and Functions" exercise into this file and save it.
Now open a new file and save it in the same directory. You should now be able to import your own module like this: importgeometry
b. Write a program to implement exception handling.

9. Write the program for the following:

a. Try to configure the widget with various options like: bg="red", family="times", size=18

b. Try to change the widget type and configuration options to experiment with other widget types like Message, Button, Entry, Checkbutton, Radiobutton, Scale etc.

10. Design the database applications for the following:

	a. Design a simple database application that stores the records and retrieve the same. b. Design a database application to search the specified record from the database. c. Design a database application to that allows the user to add, delete and modify the records.

Reference Books:

1) "Think Python" , Allen Downey, First edition, O'Reilly publication
2) "An Introduction to Computer Science using Python 3" , Jason Montojo, First Edition, SPD publication.
3) "Python GUI Programming Cookbook", Burkhard A. Meier, Packt Edition
4) "Introduction to Problem Solving with Python", E Balagurusamy , First Edition ,Tata McGraw Hill.
5) "Murach's Python Programming" , Joel Murach & Michael Urban, First Edition, SPD Publication
6) "Object Oriented Programming in Python",Michael H. Glodwasser,First Edition, Pearson Prentice Hall Publication
7) "Exploring Python", Budd, First Edition, TMH Publication

Course Code UIT3DST	**Data Structures**

Objectives

The objective of this course is to introduce the basic knowledge of algorithms and analysis procedure and determine the complexity of given algorithms and techniques.

Expected Learning Outcomes

1) Learners should be able to understand and remember algorithms and its analysis procedure.
2) Learners should be able to understand and examine the concept of data structures through ADT including List, Stack, Queue.
3) Learners should be able to compute the complexity of various algorithms.
4) Learners should be able to understand appropriate sorting/searching technique for given problem.
5) Learners should be able to design advance data structure using Non-linear data structure like tree and graph.

Unit	**Details**	**lectures**
I	**Introduction:** Data and Information, Data Structure, Classification of Data Structures, Primitive Data Types, Abstract Data Types, Data structure vs. File Organization, Operations on Data Structure, Algorithm, Importance of Algorithm Analysis, Complexity of an Algorithm, Asymptotic Analysis and Notations, Big O Notation, Big Omega Notation, Big Theta Notation, Rate of Growth and Big O Notation. **Array**: Introduction, One Dimensional Array, Memory Representation of One	**12**

	Dimensional Array, Traversing, Insertion, Deletion, Searching, Sorting, Merging of Arrays, Multidimensional Arrays, Memory Representation of Two Dimensional Arrays, General Multi-Dimensional Arrays, Sparse Arrays, Sparse Matrix, Memory Representation of Special kind of Matrices, Advantages and Limitations of Arrays.	
II	**Linked List:** Linked List, One-way Linked List, Traversal of Linked List, Searching, Memory Allocation and De-allocation, Insertion in Linked List, Deletion from Linked List, Copying a List into Other List, Merging Two Linked Lists, Splitting a List into Two Lists, Reversing One way linked List, Circular Linked List, Applications of Circular Linked List, Two way Linked List, Traversing a Two way Linked List, Searching in a Two way linked List, Insertion of an element in Two way Linked List, Deleting a node from Two way Linked List, Header Linked List, Applications of the Linked list, Representation of Polynomials, Storage of Sparse Arrays, Implementing other Data Structures.	**12**
III	**Stack**: Introduction, Operations on the Stack Memory Representation of Stack, Array Representation of Stack, Applications of Stack, Evaluation of Arithmetic Expression, Matching Parenthesis, infix and postfix operations, Recursion**.** **Queue:** Introduction, Queue, Operations on the Queue, Memory Representation of Queue, Array representation of queue, Linked List Representation of Queue, Circular Queue, Some special kinds of queues, Deque, Priority Queue, Application of Priority Queue, Applications of Queues.	**12**
IV	**Sorting and Searching Techniques** Bubble, Selection, Insertion, Merge Sort. Searching: Sequential, Binary, Indexed Sequential Searches, Binary Search. **Tree:** Tree, Binary Tree, Properties of Binary Tree, Memory Representation of Binary Tree, Operations Performed on Binary Tree, Reconstruction of Binary Tree from its Traversals, Huffman Algorithm, Binary Search Tree, Operations on Binary Search Tree, Heap, Memory Representation of Heap, Operation on Heap, Heap Sort. **Advanced Tree Structures:** Red Black Tree, Operations Performed on Red Black Tree, AVL Tree, Operations performed on AVL Tree, 2-3 Tree, B-Tree.	**12**
V	**Hashing Techniques** Hash function, Address calculation techniques, Common hashing functions Collision resolution, Linear probing, Quadratic, Double hashing, Bucket hashing, Deletion and rehashing **Graph**: Introduction, Graph, Graph Terminology, Memory Representation of Graph, Adjacency Matrix Representation of Graph, Adjacency List or Linked Representation of Graph, Operations Performed on Graph, Graph Traversal, Applications of the Graph, Reachability, Shortest Path Problems, Spanning Trees.	**12**

Course Code	**Practical List**
UIT3DSP	**1. Implement the following:** a. Write a program to store the elements in 1-D array and perform the operations like searching, sorting and reversing the elements. [Menu Driven] b. Read the two arrays from the user and merge them and display the elements in sorted order.[Menu Driven] c. Write a program to perform the Matrix addition, Multiplication and Transpose Operation. [Menu Driven] **2. Implement the following for Linked List:** a. Write a program to create a single linked list and display the node elements in reverse order.

	b. Write a program to search the elements in the linked list and display the same c. Write a program to create double linked list and sort the elements in the linked list. **3.Implement the following for Stack:** a. Write a program to implement the concept of Stack with Push, Pop, Display and Exit operations. b. Write a program to convert an infix expression to postfix and prefix conversion. c. Write a program to implement Tower of Hanoi problem. **4. Implement the following for Queue:** a. Write a program to implement the concept of Queue with Insert, Delete, Display and Exit operations. b. Write a program to implement the concept of Circular Queue c. Write a program to implement the concept of Deque. **5. Implement the following sorting techniques:** a. Write a program to implement bubble sort. b. Write a program to implement selection sort. c. Write a program to implement insertion sort. **6. Implement the following data structure techniques:** a. Write a program to implement merge sort. b. Write a program to search the element using sequential search. c. Write a program to search the element using binary search. **7. Implement the following data structure techniques:** a. Write a program to create the tree and display the elements. b. Write a program to construct the binary tree. c. Write a program for inorder, postorder and preorder traversal of tree. **8. Implement the following data structure techniques:** a. Write a program to insert the element into maximum heap. b. Write a program to insert the element into minimum heap. **9. Implement the following data structure techniques**: a. Write a program to implement the collision technique. b. Write a program to implement the concept of linear probing. **10. Implement the following data structure techniques**: a. Write a program to generate the adjacency matrix. b. Write a program for shortest path diagram.

Reference Books:

1. A Simplified Approach to Data Structures - Lalit Goyal, Vishal Goyal, Pawan Kumar
2. An Introduction to Data Structure with Applications - Jean – Paul Tremblay and Paul Sorenson.
3. Data Structure and Algorithm -Maria Rukadikar Tata McGraw Hill

Course Code UIT3CNT	**Computer Networks**

Objectives:

On completion of this course, a learner will be able to understand about computer network organization and implementation, obtaining a theoretical understanding of data communication and computer networks, and gaining practical experience in installation, monitoring, and troubleshooting of current LAN systems.

Expected Learning Outcomes:

1) Learners will be able to describe the general principles of data communication.

2) Learners will be able to describe how computer networks are organized with the concept of layered Approach. 3) Learners will be able to describe how signals are used to transfer data between nodes. 4) Learners will be able to learn about different network protocols. 5) Learners will be able to identify different network technologies		
I	**Introduction:** Data communications, networks, network types, Internet history, standards and administration. **Network Models:** Protocol layering, TCP/IP protocol suite, The OSI model. **Introduction to Physical layer:** Data and signals, periodic analog signals, digital signals, transmission impairment, data rate limits, performance. **Digital and Analog transmission:** Digital-to-digital conversion, analog-to-digital conversion, transmission modes, digital-to-analog conversion, analog-to-analog conversion.	**12**
III	**Data Link Control:** DLC services, data link layer protocols, HDLC, Point-to-point protocol. **Media Access Control**: Random access, controlled access, channelization, Wired LANs – Ethernet Protocol, standard ethernet, fast ethernet, gigabit ethernet, 10 gigabit ethernet, **Wireless LANs:** Introduction, IEEE 802.11 project, Bluetooth, WiMAX, Cellular telephony, Satellite networks. **Connecting devices and Virtual LANs.**	**12**
IV	**Introduction to the Network Layer**: Network layer services, packet switching, network layer performance, IPv4 addressing, forwarding of IP packets, Internet Protocol, ICMPv4, Mobile IP **Unicast Routing:** Introduction, routing algorithms, unicast routing protocols. **Next generation IP:** IPv6 addressing, IPv6 protocol, ICMPv6 protocol, transition from IPv4 to IPv6.	**12**
V	**Introduction to the Transport Layer**: Introduction, Transport layer protocols (Simple protocol, Stop-and-wait protocol, Go-Back-n protocol, Selective repeat protocol, Bidirectional protocols), Transport layer services, User datagram protocol, Transmission control protocol, **Standard Client0Server Protocols:** World wide-web and HTTP, FTP, Electronic mail, Telnet, Secured Shell, Domain name system.	**12**

Course Code	**Practical List**
UIT3CNP	**1. IPv4 Addressing and Subnetting** A .Given an IP address and network mask, determine other information about the IP address such as: • Network address • Network broadcast address • Total number of host bits • Number of hosts B .Given an IP address and network mask, determine other information about the IP address such as: • The subnet address of this subnet • The broadcast address of this subnet • The range of host addresses for this subnet • The maximum number of subnets for this subnet mask • The number of hosts for each subnet

	• The number of subnet bits • The number of this subnet **2.** Use of ping and tracert / traceroute, ipconfig / ifconfig, route and arp utilities. **3.** Configure IP static routing.. **4.** Configure IP routing using RIP. **5.** Configuring Simple OSPF. **6.** Configuring DHCP server and client. **7.** Create virtual PC based network using virtualization software and virtual NIC. **8.** Configuring DNS Server and client. **9.** Configuring OSPF with multiple areas. **10**.Use of Wireshark to scan and check the packet information of following protocols • HTTP • ICMP • TCP • SMTP • POP3

Reference Books:

a. "Data Communication and Networking", Behrouz A. Forouzan , Tata McGraw Hill , Fifth Edition , 2013.
b. "TCP/IP protocol suite", Behrouz A. Forouzan , Tata McGraw Hill , Fifth Edition , 2013.
c. "Computer Networks", Andrew Tanenbaum , Pearson , Fifth Edition,2013

Course Code UIT3DMS	**Advanced SQL**

Objectives

The objective of this course is to introduce the concept of the **Advanced SQL** 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.
3) Learners should be able to design relational databases.
4) Learners should be able to perform basic of PL/SQL programming.
5) Learners should be able to perform the advanced PL/SQL Programming.

I	Structured Query Language: Writing Basic SQL Select Statements, Restricting and Sorting Data, Single-Row Functions, Aggregating Data using Group Functions, Manipulating Data, Creating and Managing Tables	**12**

II	**Advanced SQL:** Subqueries, Creating Views, Creating Other Database Objects(Sequences, Indexes and Synonyms) Controlling User Access, Using SET operators, Date Time Functions, Joins (Displaying Data from Multiple Tables), **Constraints :** Constraints, types of constrains, Integrity constraints WITH Clause, Hierarchical retrieval	**12**
III	**PL-SQL:** Beginning with PL / SQL, Identifiers and Keywords, Operators, Expressions, Sequences, Overview and benefits of PL/SQL, Subprograms, types of PL/SQL blocks, Simple Anonymous Block, Identifiers, types of Identifiers, Declarative Section, variables, Scalar Data Types, The %TYPE Attribute, Executable Statements, PL/SQL Block Syntax, Comment the Code, Convert Data Types, Nested Blocks, Operators. Invoke SELECT Statements in PL/SQL, Save and Discard Transactions	**12**
IV	**Control Structures**: Conditional processing using IF Statements and CASE Statements, Loop Statement, While Loop Statement, For Loop Statement, the Continue Statement **Explicit Cursors**: Declare the Cursor, Open the Cursor, Fetch data from the Cursor, Close the Cursor, Cursor FOR loop, The %NOTFOUND and %ROWCOUNT Attributes **Exception Handling**: Handle Exceptions with PL/SQL, Propagate Exceptions Composite Type: PL/SQL Records, The %ROWTYPE Attribute, INDEX BY Tables, INDEX BY Table Methods	**12**
V	**Stored Procedures**: Create, Call, and Remove Stored Procedures, Implement Procedures Parameters and Parameters Modes **Stored Functions** Create, Call, and Remove a Stored Function, advantages of using Stored Functions, the steps to create a stored function, Invoke User-Defined Functions in SQL Statements **Packages:** advantages of Packages, components of a Package, Develop a Package, enable visibility of a Package's Components, Create the Package Specification and Body using the SQL CREATE Statement and SQL Developer, **Triggers:** the Trigger Event Types and Body, Create DML Triggers using the CREATE TRIGGER Statement, Identify the Trigger Event Types, Body, and Firing (Timing), Statement Level Triggers and Row Level Triggers, Manage, Test and Remove Triggers.	**12**

Course Code	**Practical List**
UIT3DMP	**1. Practical 1: Select queries** a. Select queries on single table using alias, where and order by clause. b. Select queries on single table using aggregate c. **2. Practical 2: Select queries using joins and unions** a. Querying data from multiple tables using all types of joins. b. Querying data from multiple tables using all types of joins. **3. Practical 3: Subqueries, DML and DDL** a. Querying single and multiple tables using subqueries.

	b. Manipulating data (Insert, update and delete) c. Creating simple tables and tables with constraints. **4. Practical 4: Creating database objects, using set operators** a. Creating Views, Sequences, Indexes and synonyms. b. Using set operators, date-time functions, **5. Practical 5: Working with advanced subqueries and WITH clause** a. Multiple column subqueries, subqueries in from clause, b. WITH Clause and hierarchical retrieval. **6. Practical 6: Basic PL/SQL, INDEX BY tables, PL/SQL Record and FOR loop.** a. Creating anonymous PL/SQL blocks. b. Define, create, and use INDEX BY tables and a PL/SQL record. **7. Practical 7: Cursors, Exceptions and procedures issuing DML and query commands.** a. Cursors with parameters to process a number of rows from multiple tables. b. Create exception handlers for specific situations. **8. Practical 8: Functions and Stored Procedures** a. Creating and invoking functions from SQL statements. b. Creating and invoking stored procedures. **9. Practical 9: Working with packages** a. Create package specifications and package bodies. Invoke the constructs in the packages. b. Create a package containing an overloaded function. c. **10.Practical 9: Working with Large Objects and triggers** a. Create statement triggers. b. Create row triggers.

Reference Books:

1) Murach's Oracle SQL and PLSQL by Joel Murach, Murach and Associates.
2) Oracle database 11g : hands on SQL/PL SQL by Satish Asnani (PHI) EEE edition
3) Programming with PL/SQL for Beginners, H. Dand, R. Patil and T. Sambare, First Edition X- Team,2011
4) PL/SQL Programming, Ivan Bayross , First Edition, BPB 2010

Course Code UIT3MAT	**Applied Mathematics**
Objectives The learners will understand the concepts of applications of the methods for solving different mathematical structures. This course introduces the advance learning of matrices and complex numbers, differential equations, Laplace transforms and the error functions.	

Expected Learning Outcomes

1) Learners should be able to solve the matrix using different methods and solve the hyperbolic functions of the complex numbers.
2) Learners should be able to identify the origin and applications of differential equations, solve initial value problems and linear DE with constant coefficient.
3) Learners should be able to understand laplace transform and apply inverse laplace to find solution of differential equations.
4) Learners should be able to find area and volume using integrals.
5) Learners should be able to find the error functions and understand the properties of beta gamma functions.

I	**Matrices:** Inverse of a matrix, Properties of matrices, Elementary Transformation, Rank of Matrix, Echelon or Normal Matrix, Inverse of matrix, Linear equations, Linear dependence and linear independence of vectors, Linear transformation, Characteristics roots and characteristics vectors, Properties of characteristic vectors, Caley-Hamilton Theorem, Similarity of matrices, Reduction of matrix to a diagonal matrix which has elements as characteristics values. **Complex Numbers:** Complex number, Equality of complex numbers, Graphical representation of complex number(Argand's Diagram), Polar form of complex numbers, Polar form of x+iy for different signs of x,y, Exponential form of complex numbers, Mathematical operation with complex numbers and their representation on Argand's Diagram, Circular functions of complex angles, Definition of hyperbolic function, Relations between circular and hyperbolic functions, Inverse hyperbolic functions, Differentiation and Integration, Graphs of the hyperbolic functions, Logarithms of complex quality, j(=i)as an operator(Electrical circuits)	**12**
II	**Equation of the first order and of the first degree:** Separation of variables, Equations homogeneous in x and y, Non-homogeneous linear equations, Exact differential Equation, Integrating Factor, Linear Equation and equation reducible to this form, Method of substitution. **Differential equation of the first order of a degree higher than the first:** Introduction, Solvable for p (or the method of factors), Solve for y, Solve for x, Clairaut's form of the equation, Methods of Substitution, Method of Substitution. **Linear Differential Equations with Constant Coefficients:** Introduction, The Differential Operator, Linear Differential Equation f(D) y = 0, Different cases depending on the nature of the root of the equation f(D) = 0, Linear differential equation f(D) y = X, The complimentary Function, The inverse operator 1/f(D) and the symbolic expiration for the particular integral 1/f(D) X; the general methods, Particular integral : Short methods, Particular integral : Other methods, Differential equations reducible to the linear differential equations with constant coefficients	**12**
III	**The Laplace Transform:** Introduction, Definition of the Laplace Transform, Table of Elementary Laplace Transforms, Theorems on Important Properties of Laplace Transformation, First Shifting Theorem, Second Shifting Theorem, The Convolution Theorem, Laplace Transform of an Integral, Laplace Transform of Derivatives, **Inverse Laplace Transform:** Shifting Theorem, Partial fraction Methods, Use of Convolution Theorem, Solution of Ordinary Linear Differential Equations with Constant Coefficients, Solution of Simultaneous Ordinary Differential Equations, Laplace Transformation of Special Function, Periodic Functions, Heaviside Unit Step Function, Dirac-delta Function(Unit Impulse Function)	**12**

IV	**Multiple Integrals:** Double Integral, Change of the order of the integration, Double integral in polar co-ordinates, Triple integrals. **Applications of integration:** Areas, Volumes of solids.	**12**
V	**Beta and Gamma Functions –** Definitions**,** Properties and Problems. Duplication formula. **Differentiation Under the Integral Sign** **Error Functions**	**12**

Books and References:

1) A text book of Applied Mathematics Vol I, P. N. Wartikar and J. N. Wartikar ,PuneVidyathi Graha
2) Applied Mathematics II , P. N. Wartikar and J. N. Wartikar ,Pune Vidyathi Graha
3) Higher Engineering Mathematics, Dr. B.S.Grewal, Khanna publications.

Course Code	**Practical List**
UIT3MPP	**List of Practical** Setting up CORDOVA, PhoneGAP Project and environment. 1. • Creating and building simple "Hello World" App using Cordova • Adding and Using Buttons • Adding and Using Event Listeners 2. • Creating and Using Functions • Using Events • Handling and Using Back Button 3. • Installing and Using Plugins • Installing and Using Battery Plugin • Installing and Using Camera Plugin 4. • Installing and Using Contacts Plugin • Installing and Using Device Plugin • Installing and Using Accelerometer Plugin 5. • Install and Using Device Orientation plugin • Install and Using Device Orientation plugin • Create and Using Prompt Function 6. • Installing and Using File Plugin • Installing and Using File Transfer Plugin • Using Download and Upload functions 7. • Installing and Using Globalization Plugin • Installing and Using Media Plugin • Installing and Using Media Capture Plugin 8. • Installing and Using Network Information Plugin

	• Installing and Using Splash Screen Plugin • Installing and Using Vibration Plugin

Reference Books:

1. Apache Cordova 4 Programming John M. Wargo 1st Addison-Wesley Professional
2. Apache Cordova in ActionRaymond Camden 1st Manning Publications

Semester IV

Course Code UIT4CJT	**Core Java**	
Objectives The objective of this course is to introduce the concept of the java programming language and understand its fundamentals. **Expected Learning Outcomes:** 1) Students will be introduced with the basic concepts of java programming. 2) Students will be able to implement java programs using control structures, iteration. 3) Students will use the advance class features including inheritance, polymorphism and overloading, overriding, interfacing, abstract classes and develop efficient and reusable codes. 4) Learners will be made familiar with multithreading, IO File handling and exception handling techniques. 5) Students will be able to design, develop and execute AWT application.		
I	**Introduction:** History, architecture and its components, Java Class File, Java Runtime Environment, The Java Virtual Machine, JVM Components, The Java API, java platform, java development kit, Lambda Expressions, Methods References, Type Annotations, Method Parameter Reflection, setting the path environment variable, Java Compiler And Interpreter, java programs, java applications, main(), public, static, void, string[] args, statements, white space, case sensitivity, identifiers, keywords, comments, braces and code blocks, variables, variable name **Data types:** primitive data types, Object Reference Types, Strings, Auto boxing, operators and properties of operators, Arithmetic operators, assignment operators, increment and decrement operator, relational operator, logical operator, bitwise operator, conditional operator.	**12**
II	**Control Flow Statements**: The If…Else If…Else Statement, The Switch…Case Statement **Iterations:** The While Loop, The Do … While Loop, The For Loop, The Foreach Loop, Labelled Statements, The Break And Continue Statements, The Return Statement **Classes:** Types of Classes, Scope Rules, Access Modifier, Instantiating Objects From A Class, Initializing The Class Object And Its Attributes, Class Methods, Accessing A Method, Method Returning A Value, Method's Arguments, Method Overloading, Variable Arguments [Varargs], Constructors, this Instance, super Instance, Characteristics Of Members Of A Class, constants, this instance, static fields of a class, static methods of a class, garbage collection	**12**
III	**Inheritance:** Derived Class Objects, Inheritance and Access Control, Default Base Class Constructors, this and super keywords. Abstract Classes And Interfaces, Abstract Classes, Abstract Methods, Interfaces, What Is An Interface? How Is An Interface Different From An Abstract Class?, Multiple	**12**

	Inheritance, Default Implementation, Adding New Functionality, Method Implementation, Classes V/s Interfaces, Defining An Interface, Implementing Interfaces. **Packages:** Creating Packages, Default Package, Importing Packages, Using A Package.	
IV	**Enumerations, Arrays:** Two Dimensional Arrays, Multi-Dimensional Arrays, Vectors, Adding Elements To A Vector, Accessing Vector Elements, Searching For Elements In A Vector, Working With The Size of The Vector. **Multithreading:** the thread control methods, thread life cycle, the main thread, creating a thread, extending the thread class. **Exceptions:** Catching Java Exceptions, Catching Run-Time Exceptions, Handling Multiple Exceptions, The finally Clause, The throws Clause **Byte streams:** reading console input, writing console output, reading file, writing file, writing binary data, reading binary data, getting started with character streams, writing file, reading file	**12**
V	**Event Handling:** Delegation Event Model, Events, Event classes, Event listener interfaces, Using delegation event model, adapter classes and inner classes. **Abstract Window Toolkit:** Window Fundamentals, Component, Container, Panel, Window, Frame, Canvas. Components – Labels, Buttons, Check Boxes, Radio Buttons, Choice Menus, Text Fields, Text, Scrolling List, Scrollbars, Panels, Frames **Layouts:** Flow Layout, Grid Layout, Border Layout, Card Layout.	**12**

Course Code	**Practical List**
UIT4CJP	**1.Java Basics** a. Write a Java program that takes a number as input and prints its multiplication table upto 10. b. Write a Java program to display the patterns. c. Write a Java program to print the area and perimeter of a circle. **2.Use of Operators** a. Write a Java program to add two binary numbers. b. Write a Java program to convert a decimal number to binary number and vice versa. c. Write a Java program to reverse a string. **3.Java Data Types** a. Write a Java program to count the letters, spaces, numbers and other characters of an input string. b. Implement a Java function that calculates the sum of digits for a given char array consisting of the digits '0' to '9'. The function should return the digit sum as a long value. c. Find the smallest and largest element from the array **4.Methods and Constructors** a. Designed a class SortData that contains the method asec() and desc(). b. Designed a class that demonstrates the use of constructor and destructor. c. Write a java program to demonstrate the implementation of abstract class. **5.Inheritance** a. Write a java program to implement single level inheritance. b. Write a java program to implement method overriding c. Write a java program to implement multiple inheritance. **6.Packages and Arrays** a. Create a package, Add the necessary classes and import the package in java class. b. Write a java program to add two matrices and print the resultant matrix. c. Write a java program for multiplying two matrices and print the product for the same. **7.Vectors and Multithreading**

	a. Write a java program to implement the vectors. b. Write a java program to implement thread life cycle. c. Write a java program to implement multithreading. **8.File Handling** a. Write a java program to open a file and display the contents in the console window. b. Write a java program to copy the contents from one file to other file. c. Write a java program to read the student data from user and store it in the file. **9.GUI and Exception Handling** a. Design a AWT program to print the factorial for an input value. b. Design an AWT program to perform various string operations like reverse string, string concatenation etc. c. Write a java program to implement exception handling. **10.GUI Programming.** a. Design an AWT application that contains the interface to add student information and display the same. b. Design a calculator based on AWT application. c. Design an AWT application to generate result marks sheet.

Reference Books:

1) "Core Java 8 for Beginners" , Vaishali Shah, Sharnam Shah, First edition, SPD publication
2) "Java: The Complete Reference" , Herbert Schildt, Ninth Edition, McGraw Hill.

Course Code UIT4EMB	**Introduction to Embedded System**	
Objectives To introduce learner with the core components of embedded system and 8051 programing in C. They will also be able to differentiate types of operating system. **Expected Learning Outcomes:** 1) Become familiar with classification, characteristics, core components of embedded system 2) Become familiar with memory, types of memory, registers 3) Acquire skills in 8051 programing in C 4) Acquire skills for selecting microcontroller and developing basic applications 5) Become familiar with different types of operating system and its characteristics.		
I	**Introduction:** Embedded Systems and general purpose computer systems, history, classifications, applications and purpose of embedded systems **Core of embedded systems:** microprocessors and microcontrollers, RISC and CISC controllers, Big endian and Little endian processors, Application specific ICs, Programmable logic devices, COTS, sensors and actuators, communication interface, embedded firmware, other system components. **Characteristics and quality attributes of embedded systems:** Characteristics, operational and non-operational quality attributes.	**12**
II	**Embedded Systems – Application and Domain Specific:** Application specific – washing machine, domain specific - automotive. **Embedded Hardware:** Memory map, i/o map, interrupt map, processor family, external peripherals, memory – RAM , ROM, types of RAM and ROM, memory testing, CRC	**12**

	,Flash memory. **Peripherals:** Control and Status Registers, Device Driver, Timer Driver - Watchdog Timer	
III	**The 8051 Microcontrollers:** Microcontrollers and Embedded processors, Overview of 8051 family.8051 Microcontroller hardware, Input/output pins, Ports, and Circuits, External Memory. **8051 Programming in C:** Data Types and time delay in 8051 C, I/O Programming, Logic operations, Data conversion Programs.	**12**
IV	**Designing Embedded System with 8051 Microcontroller:** Factors to be considered in selecting a controller, why 8051 Microcontroller, Designing with 8051. **Programming embedded systems:** structure of embedded program, infinite loop, compiling, linking and debugging	**12**
V	**Real Time Operating System (RTOS):** Operating system basics, types of operating systems, Real-Time Characteristics, Selection Process of an RTOS. **Design and Development:** Embedded system development Environment – IDE, types of file generated on cross compilation, disassembler/ de-compiler, simulator, emulator and debugging, embedded product development life-cycle, trends in embedded industry.	**12**

Course Code	**Practical List**
UIT4ESP	**1)Design and develop a reprogrammable embedded computer using 8051 microcontrollers and to show the following aspects**. a. Programming b. Execution c. Debugging 2A) Configure timer control registers of 8051 and develop a program to generate given time delay. 2B) To demonstrate use of general purpose port i.e. Input/ output port of two controllers for data transfer between them. 3A) Port I / O: Use one of the four ports of 8051 for O/P interfaced to eight LED's. Simulate binary counter (8 bit) on LED's 3B) To interface 8 LEDs at Input-output port and create different patterns. 3C) To demonstrate timer working in timer mode and blink LED without using any loop delay routine. 4A) Serial I / O: Configure 8051 serial port for asynchronous serial communication with serial port of PC exchange text messages to PC and display on PC screen. Signify end of message by carriage return. 4B) To demonstrate interfacing of seven-segment LED display and generate counting from 0 to 99 with fixed time delay. 4C) Interface 8051 with D/A converter and generate square wave of given frequency on oscilloscope.

	5A) Interface 8051 with D/A converter and generate triangular wave of given frequency on oscilloscope. 5B) Using D/A converter generate sine wave on oscilloscope with the help of lookup table stored in data area of 8051. 6) Interface stepper motor with 8051 and write a program to move the motor through a given angle in clock wise or counter clock wise direction. 7) Generate traffic signal. 8) Implement Temperature controller. 9) Implement Elevator control. 10) Using Flash Magic a. To demonstrate the procedure for flash programming for reprogrammable embedded system board using Flash Magic b. To demonstrate the procedure and connections for multiple controllers programming of same type of controller with same source code in one go, using flash magic.

Reference Books:

1) Introduction to embedded systems ,Shibu K V, First, Tata Mcgraw-Hill
2) Embedded Systems Rajkamal Tata Mcgraw-Hill

Course Code UIT4COS	**Computer Oriented Statistical Techniques**

Objectives

The objective of this course is to provide an understanding for the learners on statistical concepts to include measures of dispersion probability distribution, sampling estimation, hypothesis testing, regression and correlation analysis.

Expected Learning Outcomes:

By completing this course the learners will be able to perform:

1. To calculate and apply measures of dispersion.
2. To apply discrete and continuous probability distribution to various problems.
3. The test of hypothesis as well as calculate confidence interval for a population parameter and learn the concept to p-value.
4. Learn non parametric test such as the Chi- Square test for Independence as well as goodness of fit.
5. to compute and interpret the results of bivariate and multivariate regression and correlation analysis and to perform ANOVA. Be able to perform multiple regression using computer software R.

I		**12**

	The Mean, Median, Mode, and Other Measures of Central Tendency: Index, or Subscript, Notation, Summation Notation, Averages, or Measures of Central Tendency ,The Arithmetic Mean , The Weighted Arithmetic Mean ,Properties of the Arithmetic Mean ,The Arithmetic Mean Computed from Grouped Data ,The Median ,The Mode, The Empirical Relation Between the Mean, Median, and Mode, The Geometric Mean G, The Harmonic Mean H ,The Relation Between the Arithmetic, Geometric, and Harmonic Means, The Root Mean Square, Quartiles, Deciles, and Percentiles, Software and Measures of Central Tendency. **The Standard Deviation and Other Measures of Dispersion**: Dispersion, or Variation, The Range, The Mean Deviation, The Semi-Interquartile Range, The 10–90 Percentile Range, The Standard Deviation, The Variance, Short Methods for Computing the Standard Deviation, Properties of the Standard Deviation, Charlie's Check, Sheppard's Correction for Variance, Empirical Relations Between Measures of Dispersion, Absolute and Relative Dispersion; Coefficient of Variation, Standardized Variable; Standard Scores, Software and Measures of Dispersion. **Introduction to R:** Basic syntax, data types, variables, operators, control statements, R-functions, R –Vectors, R – lists, R Arrays .	
II	**Moments, Skewness, and Kurtosis** : Moments , Moments for Grouped Data ,Relations Between Moments , Computation of Moments for Grouped Data, Charlie's Check and Sheppard's Corrections, Moments in Dimensionless Form, Skewness, Kurtosis, Population Moments, Skewness, and Kurtosis, Software Computation of Skewness and Kurtosis. **Elementary Probability Theory**: Definitions of Probability, Conditional Probability; Independent and Dependent Events, Mutually Exclusive Events, Probability Distributions, Mathematical Expectation, Relation Between Population, Sample Mean, and Variance, Combinatorial Analysis, Combinations, Stirling's Approximation to n!, Relation of Probability to Point Set Theory, Euler or Venn Diagrams and Probability. **Elementary Sampling Theory** : Sampling Theory, Random Samples and Random Numbers, Sampling With and Without Replacement, Sampling Distributions, Sampling Distribution of Means, Sampling Distribution of Proportions, Sampling Distributions of Differences and Sums, Standard Errors, Software Demonstration of Elementary Sampling Theory .	**12**
III	**Statistical Estimation Theory:** Estimation of Parameters, Unbiased Estimates, Efficient Estimates, Point Estimates and Interval Estimates; Their Reliability, Confidence-Interval Estimates of Population Parameters, Probable Error. **Statistical Decision Theory:** Statistical Decisions, Statistical Hypotheses, Tests of Hypotheses and Significance, or Decision Rules, Type I and Type II Errors, Level of Significance, Tests Involving Normal Distributions, Two-Tailed and One-Tailed Tests, Special Tests, Operating-Characteristic Curves; the Power of a Test, p-Values for Hypotheses Tests, Control Charts, Tests Involving Sample Differences, Tests Involving Binomial Distributions. **Statistics in R:** Mean, Median, Mode, Normal Distribution , Binomial Distribution, Frequency Distribution in R	**12**
IV	**Small Sampling Theory**: Small Samples, Student's t Distribution, Confidence Intervals, Tests of Hypotheses and Significance, The Chi-Square Distribution, Confidence Intervals for Sigma , Degrees of Freedom, The F Distribution. **The Chi-Square Test**: Observed and Theoretical Frequencies, Definition of chi-square, Significance Tests, The Chi-Square Test for Goodness of Fit, Contingency Tables, Yates' Correction for Continuity, Simple Formulas for Computing chi-square, Coefficient of Contingency, Correlation of Attributes, Additive Property of chi-square	12

V	**Curve Fitting and the Method of Least Squares:** Relationship Between Variables, Curve Fitting, Equations of Approximating Curves, Freehand Method of Curve Fitting, The Straight Line, The Method of Least Squares, The Least-Squares Line, Nonlinear Relationships, The Least-Squares Parabola, Regression, Applications to Time Series, Problems Involving More Than Two Variables. **Correlation Theory:** Correlation and Regression, Linear Correlation, Measures of Correlation, The Least-Squares Regression Lines, Standard Error of Estimate, Explained and Unexplained Variation, Coefficient of Correlation, Remarks Concerning the Correlation Coefficient, Product-Moment Formula for the Linear Correlation Coefficient, Short Computational Formulas, Regression Lines and the Linear Correlation Coefficient, Correlation of Time Series, Correlation of Attributes, Sampling Theory of Correlation, Sampling Theory of Regression	**12**

Course Code	**Practical List**
UIT4COP	1.Using R execute the basic commands, array, list and frames. 2.Create a Matrix using R and Perform the operations addition, inverse, transpose and multiplication operations. 3.Using R Execute the statistical functions: mean, median, mode, quartiles, range, inter quartile range histogram 4.Using R import the data from Excel / .CSV file and Perform the above functions. 5.Using R import the data from Excel / .CSV file and Calculate the standard deviation, variance, co-variance. 6. Using R import the data from Excel / .CSV file and draw the skewness. 7. Import the data from Excel / .CSV and perform the hypothetical testing. 8. Import the data from Excel / .CSV and perform the Chi-squared Test. 9.Using R perform the binomial and normal distribution on the data. 10.Perform the Linear Regression using R. 11.Compute the Least squares means using R. 12.Compute the Linear Least Square Regression

Reference Books:

1. STATISTICS, Murray R Spiegel, Larry J. Stephens, McGRAW –HILL ITERNATIONAL, Fourth edition.
2. FUNDAMENTAL OF MATHEMATICAL STATISTICS S.C. GUPTA and V.K. KAPOOR, SULTAN CHAND and SONS, ELEVENTH EDITION.
3. A Practical Approach using R , R.B. Patil, H.J. Dand and R. Bhavsar , SPD publication, First edition.

Course Code UIT4SWE	**Software Engineering**

Objectives

1. Basic knowledge and understanding of the analysis and design of complex systems.
2. Ability to apply software engineering principles and techniques.
3. Ability to develop, maintain and evaluate large-scale software systems.
4. To produce efficient, reliable, robust and cost-effective software solutions.

Expected Learning Outcomes:

1. Ability to perform independent research and analysis.
2. To communicate and coordinate competently by listening, speaking, reading and writing English for technical and general purposes.

3. Ability to work as an effective member or leader of software engineering teams.
4. To manage time, processes and resources effectively by prioritizing competing demands to achieve personal and team goals Identify and analyzes the common threats in each domain.
5. Ability to understand and meet ethical standards and legal responsibilities.

I	**Introduction:** What is software engineering? Software Development Life Cycle, Requirements Analysis, Software Design, Coding, Testing, Maintenance etc. Software Requirements: Functional and Non-functional requirements, User Requirements, System Requirements, Interface Specification, Documentation of the software requirements. **Software Processes:** Process and Project, Component Software Processes. **Software Development Process Models.** • Waterfall Model. • Prototyping. • Iterative Development. • Rational Unified Process. • The RAD Model • Time boxing Model. **Agile software development**: Agile methods, Plan-driven and agile development, Extreme programming, Agile project management, Scaling agile methods.	**12**
II	**Socio-technical system:** Essential characteristics of socio technical systems, Emergent System Properties, Systems Engineering, Components of system such as organization, people and computers, Dealing Legacy Systems. **Critical system:** Types of critical system, A simple safety critical system, Dependability of a system, Availability and Reliability, Safety and Security of Software systems. **Requirements Engineering Processes:** Feasibility study, Requirements elicitation and analysis, Requirements Validations, Requirements Management. **System Models:** Models and its types, Context Models, Behavioural Models, Data Models, Object Models, Structured Methods.	**12**
III	**Architectural Design:** Architectural Design Decisions, System Organisation, Modular Decomposition Styles, Control Styles, Reference Architectures. **User Interface Design:** Need of UI design, Design issues, The UI design Process, User analysis, User Interface Prototyping, Interface Evaluation. **Project Management:** Software Project Management, Management activities, Project Planning, Project Scheduling, and Risk Management. **Quality Management:** Process and Product Quality, Quality assurance and Standards, Quality Planning, Quality Control, Software Measurement and Metrics.	**12**
IV	**Verification and Validation:** Planning Verification and Validation, Software Inspections, Automated Static Analysis, Verification and Formal Methods. **Software Testing:** System Testing, Component Testing, Test Case Design, Test Automation. **Software Measurement**: Size-Oriented Metrics, Function-Oriented Metrics, Extended Function Point Metrics **Software Cost Estimation:** Software Productivity, Estimation Techniques, Algorithmic Cost Modelling, Project Duration and Staffing	**12**
V	**Process Improvement:** Process and product quality, Process Classification, Process Measurement, Process Analysis and Modelling, Process Change, The CMMI Process Improvement Framework. **Service Oriented Software Engineering:** Services as reusable components, Service Engineering, Software Development with Services. **Software reuse:** The reuse landscape, Application frameworks, Software product lines,	**12**

	COTS product reuse. **Distributed software engineering**: Distributed systems issues, Client–server computing, Architectural patterns for distributed systems, Software as a service.	

Course Code	**Practical List**
UIT4SEP	1. Study and implementation of class diagrams. 2. Study and implementation of Use Case Diagrams. 3. Study and implementation of Entity Relationship Diagrams. 4. Study and implementation of Sequence Diagrams. 5. Study and implementation of State Transition Diagrams. 6. Study and implementation of Data Flow Diagrams. 7. Study and implementation of Collaboration Diagrams. 8. Study and implementation of Activity Diagrams. 9. Study and implementation of Component Diagrams. 10. Study and implementation of Deployment Diagrams.

Reference Books:

1. Software Engineering, edition, Ian Somerville Pearson Education. Edition Ninth
2. Software Engineering Pankaj Jalote Narosa Publication
3. Software engineering, a practitioner's approach Roger Pressman Tata Mcgraw-hill Seventh edition
4. Software Engineering principles and practice WS Jawadekar Tata Mcgraw-hill

Course Code UIT4CGA	**Computer Graphics and Animation**	
Objectives To make the learners aware of the different algorithms that are actually used for developing different types of animations. This subject aims in making the learners capable of designing different animations programmatically. **Expected Learning Outcomes:** 1. Learners will know the different algorithms used for computer graphics. 2. Learners will gain knowledge about projections and transformations. 3. Acquire basic knowledge about 3D effects 4. Learners will know different types of surfaces and their basic algorithms. 5. Understand Animation and storage processing.		
I	**Introduction to Computer Graphics:** Overview of Computer Graphics, Computer Graphics Application and Software, Description of some graphics devices, Input Devices for Operator Interaction, Active and Passive Graphics Devices, Display Technologies, Storage Tube Graphics Displays, Calligraphic Refresh Graphics Displays, Raster Refresh (Raster-Scan) Graphics Displays, Cathode Ray Tube Basics, Color CRT Raster Scan Basics, Video Basics, The Video Controller, Random-Scan Display Processor, LCD displays. **Scan conversion** – Digital Differential Analyzer (DDA) algorithm, Bresenhams' Line drawing algorithm. Bresenhams' method of Circle drawing, Midpoint Circle Algorithm, Midpoint Ellipse Algorithm, Mid-point criteria, Problems of Aliasing, end-point ordering and clipping lines, Scan Converting Circles, Clipping Lines algorithms– Cyrus-Beck,	**12**

	Cohen-Sutherland and Liang-Barsky, Clipping Polygons, problem with multiple components. • Time boxing Model. **Agile software development**: Agile methods, Plan-driven and agile development, Extreme programming, Agile project management, Scaling agile methods.	
II	**Two-Dimensional Transformations:** Transformations and Matrices, Transformation Conventions, 2D Transformations, Homogeneous Coordinates and Matrix Representation of 2D Transformations, Translations and Homogeneous Coordinates, Rotation, Reflection, Scaling, Combined Transformation, Transformation of Points, Transformation of The Unit Square, Solid Body Transformations, Rotation About an Arbitrary Point, Reflection through an Arbitrary Line, A Geometric Interpretation of Homogeneous Coordinates, The Window-to-Viewport Transformations. **Three-Dimensional Transformations:** Three-Dimensional Scaling, Three-Dimensional Shearing, Three-Dimensional Rotation, Three-Dimensional Reflection, Three-Dimensional Translation, Multiple Transformation, Rotation about an Arbitrary Axis in Space, Reflection through an Arbitrary Plane, Matrix Representation of 3D Transformations, Composition of 3D Transformations, Affine and Perspective Geometry, Perspective Transformations, Techniques for Generating Perspective Views, Vanishing Points, the Perspective Geometry and camera models, Orthographic Projections, Axonometric Projections, Oblique Projections, View volumes for projections.	**12**
III	**Viewing in 3D** Stages in 3D viewing, Canonical View Volume (CVV), Specifying an Arbitrary 3D View, Examples of 3D Viewing, The Mathematics of Planar Geometric Projections, Combined transformation matrices for projections and viewing, Coordinate Systems and matrices, camera model and viewing pyramid. **Light:** Radiometry, Transport, Equation, Photometry **Color:** Colorimetry, Color Spaces, Chromatic Adaptation, Color Appearance	**12**
IV	**Visible-Surface Determination:** Techniques for efficient Visible-Surface Algorithms, Categories of algorithms, Back face removal, The z-Buffer Algorithm, Scan-line method, Painter's algorithms (depth sorting), Area sub-division method, BSP trees, Visible-Surface Ray Tracing, comparison of the methods. **Plane Curves and Surfaces:** Curve Representation, Nonparametric Curves, Parametric Curves, Parametric Representation of a Circle, Parametric Representation of an Ellipse, Parametric Representation of a Parabola, Parametric Representation of a Hyperbola, Representation of Space Curves, Cubic Splines, , Bezier Curves, B-spline Curves, B-spline Curve Fit, B-spline Curve Subdivision, Parametric Cubic Curves, Quadric Surfaces. Bezier Surfaces.	**12**
V	**Computer Animation:** Principles of Animation, Key framing, Deformations, Character Animation, Physics-Based Animation, Procedural Techniques, Groups of Objects. **Image Manipulation and Storage:** What is an Image? Digital image file formats, Image compression standard – JPEG, Image Processing - Digital image enhancement, contrast stretching, Histogram Equalization, smoothing and median Filtering.	**12**

Course Code	**Practical List**

UIT4CGP	**1. Solve the following:** a. Study and enlist the basic functions used for graphics in C / C++ / Python language. Give an example for each of them. Draw a co-ordinate axis at the center of the screen. **2. Solve the following**: a. Divide your screen into four region, draw circle, rectangle, ellipse and half ellipse in each regionwith appropriate message. b. Draw a simple hut on the screen. **3. Draw the following basic shapes in the center of the screen :** i. Circle ii. Rectangle iii. Square iv. Concentric Circles v. Ellipse vi. Line **4. Solve the following:** a. Develop the program for DDA Line drawing algorithm. b. Develop the program for Bresenham's Line drawing algorithm. **5. Solve the following:** a. Develop the program for the mid-point circle drawing algorithm. b. Develop the program for the mid-point ellipse drawing algorithm. **6. Solve the following:** a. Write a program to implement 2D scaling. b. Write a program to perform 2D translation **7. Solve the following:** a. Perform 2D Rotation on a given object. b. Program to create a house like figure and perform the following operations. i. Scaling about the origin followed by translation. ii. Scaling with reference to an arbitrary point. iii. Reflect about the line y = mx + c. **8. Solve the following:** a.Write a program to implement Cohen-Sutherland clipping. b. Write a program to implement Liang - Barsky Line Clipping Algorithm **9. Solve the following:** a. Write a program to fill a circle using Flood Fill Algorithm. b. Write a program to fill a circle using Boundary Fill Algorithm. **10. Solve the following:** a.Develop a simple text screen saver using graphics functions. b.Perform smiling face animation using graphic functions. c.Draw the moving car on the screen.

Reference Books:

1. Computer Graphics - Principles and Practice J. D. Foley, A. Van Dam, S. K. Feiner and J. F. Hughes Pearson 2nd edition.
2. Computer Graphics Hearn, Baker Pearson 2nd edition.
3. Fundamentals of Computer Graphics Steve Marschner, Peter Shirley 4th edition.

JanardanBhagatShikshanPrasarakSanstha'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 T.Y.B.Sc. Information Technology
Choice Based Credit & Grading System (60:40)
w.e.f. Academic Year 2021-2022

Sr. No.	Heading	Particulars
1	Title of Course	Information Technology
2	Eligibility for Admission	(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. (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. (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. OR 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.
3	Passing marks	40%

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	2021-22

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

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

1) Learners are able to work effectively in IT industries in field of project management.
2) Able to realize the revolution of Internet in Mobile Devices, Cloud & Sensor Networks. Able to understand building blocks of Internet of Things and characteristics.
3) Learners are able to configure different types of servers on Linux Platform.
4) Learners are able to create application projects using different technologies such as enterprise java and .Net.
5) Learners are able to build and enhance business intelligence capabilities by adapting the appropriate technology and software solutions.

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 PatternforClass 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)
T.Y.B. Sc. Information Technology Syllabus
To be implemented from the Academic year 2021-2022
SEMESTER V

Course Code	Course Type	Course Title	Credits
UIT5SPM	Skill Enhancement Course	Software Project Management	2
UIT5IOT	Skill Enhancement Course	Internet of Things	2
UIT5AWP	Skill Enhancement Course	Advanced Web Programming	2
UIT5LSA	Discipline Specific	Linux System Administration	2
UIT5ENJ	Discipline Specific	Enterprise Java	2
UIT5PDP	Skill Enhancement Course Practical	Project Dissertation	2
UIT5ITP	Skill Enhancement Course Practical	Internet of Things Practical	2
UIT5WPP	Skill Enhancement Course Practical	Advanced Web Programming Practical	2
UIT5LAP	Discipline Specific Practical	Linux Administration Practical	2
UIT5EJP	Discipline Specific Practical	Enterprise Java Practical	2
Total Credits		**20**	

Choice Based Credit Grading and Semester System (CBCGS)
T.Y.B. Sc. Information Technology Syllabus
To be implemented from the Academic year 2021-2022
SEMESTER VI

Course Code	Course Type	Course Title	Credits
UIT6SQA	Skill Enhancement Course	Software Quality Assurance	2
UIT6SIC	Skill Enhancement Course	Security in Computing	2
UIT6BUI	Skill Enhancement Course	Business Intelligence	2
UIT6GIS	Discipline Specific Elective	Principles of Geographic Information Systems	2
UIT6ISM	Discipline Specific Elective	IT Service Management	2
UIT6PIP	Skill Enhancement Course Practical	Project Implementation	2
UIT6SCP	Skill Enhancement	Security in	2

	Course Practical	Computing Practical	
UIT6BI P	Skill Enhancement Course Practical	Business Intelligence Practical	2
UIT6GIP	Discipline Specific Elective Practical	Principles of Geographic Information Systems Practical	2
UIT6ISP	Skill Enhancement Course Practical	Advanced Mobile Programming	2
Total Credits		**20**	

Semester V

Course Code UIT5SPM	**Software Project Management**

Objectives

The objective of this course is to understand some problem and concern of software project manager, learners will able to cost estimation of project.

Expected Learning Outcomes:

1) Learners will be able to clear the idea about project planning.
2) Learners will be able to determine Success criteria for a project.
3) Learners will be able to reduce some risk certain of appropriate prototype
4) Learners will be able to determine, estimate the overall duration of project.
5) Learners will be able to Identify the resource requirements.

I	**Introduction to Software Project Management**: Introduction, Why is Software Project Management Important? What is a Project? Software Projects versus Other Types of Project, Contract Management and Technical Project Management, Activities Covered by Software Project Management, Plans, Methods and Methodologies, Some Ways of Categorizing Software Projects, Project Charter, Stakeholders, Setting Objectives, The Business Case, Project Success and Failure, What is Management? Management Control, Project Management Life Cycle, Traditional versus Modern Project Management Practices. **Project Evaluation and Programme Management**: Introduction, Business Case, Project Portfolio Management, Evaluation of Individual Projects, Cost–benefit Evaluation Techniques, Risk Evaluation, Programme Management, Managing the Allocation of Resources within Programmes, Strategic Programme Management, Creating a Programme, Aids to Programme Management, Some Reservations about Programme Management, Benefits Management. **An Overview of Project Planning** : Introduction to Step Wise Project Planning, Step 0: Select Project, Step 1: Identify Project Scope and Objectives, Step 2: Identify Project Infrastructure, Step 3: Analyse Project Characteristics, Step 4: Identify Project Products and Activities, Step 5: Estimate Effort for Each Activity, Step 6: Identify Activity Risks, Step 7: Allocate Resources, Step 8: Review/Publicize Plan, Steps 9 and 10: Execute Plan/Lower Levels of Planning	**12**
II	**Selection of an Appropriate Project Approach:** Introduction, Build or Buy? Choosing Methodologies and Technologies, Software Processes and Process Models, Choice of Process Models, Structure versus Speed of Delivery, The Waterfall Model, The Spiral Model, Software Prototyping, Other Ways of Categorizing Prototypes, Incremental Delivery, Atern/Dynamic Systems Development Method, Rapid Application Development, Agile Methods, Extreme Programming (XP), Scrum, Lean Software Development, Managing Iterative Processes, Selecting the Most Appropriate Process Model. **Software Effort Estimation:** Introduction, Where are the Estimates Done? Problems with Over- and Under-Estimates, The Basis for Software Estimating, Software Effort Estimation Techniques, Bottom-up Estimating, The Top-down Approach and Parametric Models, Expert Judgement, Estimating by Analogy, Albrecht Function Point Analysis, Function Points Mark II, COSMIC Full Function Points, COCOMO II: A Parametric Productivity Model, Cost Estimation, Staffing Pattern, Effect of Schedule Compression, Capers Jones Estimating Rules of Thumb.	**12**

III	**Activity Planning:** Introduction, Objectives of Activity Planning, When to Plan, Project Schedules, Projects and Activities, Sequencing and Scheduling Activities, Network Planning Models, Formulating a Network Model, Adding the Time Dimension, The Forward Pass, Backward Pass, Identifying the Critical Path, Activity Float, Shortening the Project Duration, Identifying Critical Activities, Activity-on-Arrow Networks. **Risk Management**: Introduction, Risk, Categories of Risk, Risk Management Approaches, A Framework for Dealing with Risk, Risk Identification, Risk Assessment, Risk Planning, Risk Management, Evaluating Risks to the Schedule, Boehm's Top 10 Risks and Counter Measures, Applying the PERT Technique, Monte Carlo Simulation, Critical Chain Concepts. **Resource Allocation**: Introduction, Nature of Resources, Identifying Resource Requirements, Scheduling Resources, Creating Critical Paths, Counting the Cost, Being Specific, Publishing the Resource Schedule, Cost Schedules, Scheduling Sequence.	**12**
IV	**Monitoring and Control**: Introduction, Creating the Framework, Collecting the Data, Review, Visualizing Progress, Cost Monitoring, Earned Value Analysis, Prioritizing Monitoring, Getting the Project Back to Target, Change Control, Software Configuration Management (SCM). **Managing Contracts**: Introduction, Types of Contract, Stages in Contract Placement, Typical Terms of a Contract, Contract Management, Acceptance. **Managing People in Software Environments**: Introduction, Understanding Behaviour, Organizational Behaviour: A Background, Selecting the Right Person for the Job, Instruction in the Best Methods, Motivation, The Oldham–Hackman Job Characteristics Model, Stress, Stress Management, Health and Safety, Some Ethical and Professional Concerns.	**12**
V	**Working in Teams:** Introduction, Becoming a Team, Decision Making, Organization and Team Structures, Coordination Dependencies, Dispersed and Virtual Teams, Communication Genres, Communication Plans, Leadership. **Software Quality :**Introduction, The Place of Software Quality in Project Planning, Importance of Software Quality, Defining Software Quality, Software Quality Models, ISO 9126, Product and Process Metrics, Product versus Process Quality Management, Quality Management Systems, Process Capability Models, Techniques to Help Enhance Software Quality, Testing, Software Reliability, Quality Plans.	**12**

Course Code	**Project Dessertation and Viva**
UIT5PDP	Chapter 1 to 4 should be submitted in Semester V in spiral binding. These chapter have also to be included in Semester VI report. Semester VI report has to be hard bound with golden embossing. Students will be evaluated based on the dissertation in semester V and dissertation and viva voce in Semester VI. **I. OBJECTIVES** **II. Type of the Project** **III. SOFTWARE AND BROAD AREAS OF APPLICATION FRONT END** **IV. Introduction** **1.1 PROJECT REPORT:** **CHAPTER 1: INTRODUCTION** **CHAPTER 2: SURVEY OF TECHNOLOGIES** **CHAPTER 3: REQUIREMENTS AND ANALYSIS**

	CHAPTER 4: SYSTEM DESIGN **CHAPTER 5: IMPLEMENTATION AND TESTING** **CHAPTER 6: RESULTS AND DISCUSSION** **V. EXPLANATION OF CONTENTS** **The details are given in Appendix–I**

ReferenceBooks:

1) Software Project Management, Bob Hughes, Mike Cotterell, Rajib Mall sixth edition.
2) Project Management and Tools & Technologies – An overview,Shailesh Mehta 1st edition

Course Code UIT5IOT	**Internet of Things**	
Objectives: To assess the vision and introduction of IoT. To Understand IoT Market perspective. To provide an understanding of the technologies and the standards relating to the Internet of Things. **Expected Learning Outcomes** 1) Learners will be able to Interpret the vision of IoT from a global context 2)Learners will be able to become familiar with IoT hardware components 3)Learners will be able to acquire skills to design 3D modules 4)Learners will be able to determine the Market perspective of IoT 5)Learners will be able to acquire skills on developing their enterprise level technical strategies		
I	**The Internet of Things: An Overview** : The Flavour of the Internet of Things, The "Internet" of "Things", The Technology of the Internet of Things, Enchanted Objects, Who is Making the Internet of Things? **Design Principles for Connected Devices:** Calm and Ambient Technology, Magic as Metaphor, Privacy, Keeping Secrets, Whose Data Is It Anyway? Web Thinking for Connected Devices, Small Pieces, Loosely Joined, First-Class Citizens On The Internet, Graceful Degradation, Affordances. **Internet Principles:** Internet Communications: An Overview, IP, TCP, The IP Protocol Suite (TCP/IP), UDP, IP Addresses, DNS, Static IP Address Assignment, Dynamic IP Address Assignment, IPv6, MAC Addresses, TCP and UDP Ports, An Example: HTTP Ports, Other Common Ports, Application Layer Protocols, HTTP, HTTPS: Encrypted HTTP, Other Application Layer Protocols.	**12**
II	**Thinking About Prototyping:** Sketching, Familiarity, Costs versus Ease of Prototyping, Prototypes and Production, Changing Embedded Platform, Physical Prototypes and Mass Personalisation, Climbing into the Cloud, Open Source versus Closed Source, Why Closed? Why Open? Mixing Open and Closed Source, Closed Source for Mass Market Projects, Tapping into the Community. **Prototyping Embedded Devices:** Electronics, Sensors, Actuators, Scaling Up the Electronics, Embedded Computing Basics, Microcontrollers, System-on-Chips, Choosing Your Platform, Arduino, Developing on the Arduino, Some Notes on the Hardware, Openness, Raspberry Pi, Cases and Extension Boards, Developing on the Raspberry Pi, Some Notes on the Hardware, Openness.	**12**
III	**Prototyping the Physical Design:** Preparation, Sketch, Iterate, and Explore, Nondigital Methods, Laser Cutting, Choosing a Laser Cutter, Software, Hinges and Joints, 3D Printing, Types of 3D Printing, Software, CNC Milling, Repurposing/Recycling. **Prototyping Online Components:** Getting Started with an API, Mashing Up APIs, Scraping, Legalities, Writing a New API, Clockodillo, Security, Implementing the API, Using Curl to Test, Going Further, Real-Time Reactions, Polling, Comet, Other Protocols, MQ Telemetry Transport,	**12** **12**
IV	**Techniques for Writing Embedded Code:** Memory Management, Types of Memory, Making the Most of Your RAM, Performance and Battery Life, Libraries, Debugging, **Business Models:** A Short History of Business Models, Space and Time, From Craft to Mass Production, The Long Tail of the Internet, Learning from History, The Business Model Canvas, Who Is the Business Model For? Models, Make Thing, Sell Thing, Subscriptions, Customisation, Be a Key Resource, Provide Infrastructure: Sensor	**12**

	Networks, Take a Percentage, Funding an Internet of Things Startup, Hobby Projects and Open Source, Venture Capital, Government Funding, Crowdfunding, Lean Startups.	
V	**Moving to Manufacture:** What Are You Producing? Designing Kits, Designing Printed circuit boards, Software Choices, The Design Process, Manufacturing Printed Circuit Boards, Etching Boards, Milling Boards. Assembly, Testing, Mass-Producing the Case and Other Fixtures, Certification, Costs, Scaling Up Software, Deployment, Correctness and Maintainability, Security, Performance, User Community. **Ethics:** Characterizing the Internet of Things, Privacy, Control, Disrupting Control, Crowdsourcing, Environment, Physical Thing, Electronics, Internet Service, Solutions, The Internet of Things as Part of the Solution, Cautious Optimism, The Open Internet of Things Definition.	**12**

Course Code	**Internet of Things Practical List**
UIT5ITP	1 Starting Raspbian OS, Familiarising with Raspberry Pi Components and interface, Connecting to Ethernet , Monitor and USB **2** Displaying different LED patterns with Raspberry Pi. **3** Displaying Time over 4-Digit 7-Segment Display using Raspberry Pi **4** Raspberry Pi Based Oscilloscope **5** Controlling Raspberry Pi with WhatsApp. **6** Setting up Wireless Access Point using Raspberry Pi **7** Fingerprint Sensor interfacing with Raspberry Pi **8** Raspberry Pi GPS Module Interfacing **9** IoT based Web Controlled Home Automation using Raspberry Pi **10** Visitor Monitoring with Raspberry Pi and Pi Camera **11** Interfacing Raspberry Pi with RFID. **12** Building Google Assistant with Raspberry Pi. **13** Installing Windows 10 IoT Core on Raspberry Pi

Reference Books:

1) Designing the Internet of Things , "Adrian McEwen" First Edition,WILEY
2) Internet of Things – Architecture and Design, "Raj Kamal", First Edition,McGraw Hill

Course Code	**Advanced Web Programming**
UIT5AWP	

Objectives: To learn to develop web applications that use object-oriented techniques and advanced database interactions. Concepts such as advanced CSS concepts web environment, authentication and security will also be explored.

Expected Learning Outcomes:

1) Learners will be able to do programming with C# Language.

	2) Learners will be able to acquire skills to design web page incorporate with different server controls on webpages. 3) Learners will be able to acquire skills to handle Error Handling, Logging, and Tracing , State Management 4) Learners will be able to acquire skills to develop dynamic web pages using ADO.NET Fundamentals 5) Learners will be able to provide interaction between web pages using ASP.NET AJAX.	
I	**Introducing .NET**: The .NET Framework, C#, VB, and the .NET Languages, The Common Language Runtime, The .NET Class Library. .NET programming tools: Visual Studio Gallery ,LINQPad , SQL Complete. **The C# Language**: C# Language Basics, Variables and Data Types, Variable Operations, Object-Based Manipulation, Conditional Logic, Loops, Methods. **Types, Objects, and Namespaces**: The Basics About Classes, Building a Basic Class, Value Types and Reference Types, Understanding Namespaces and Assemblies, Advanced Class Programming.	**12**
II	**Web Form Fundamentals**: Writing Code, Using the Code-Behind Class, Adding Event Handlers, Understanding the Anatomy of an ASP.NET Application, Introducing Server Controls, Using the Page Class, Using Application Events, Configuring an ASP.NET Application. **Form Controls**: Stepping Up to Web Controls, Web Control Classes, List Controls, Table Controls, Web Control Events and AutoPostBack, Validation, Understanding Validation, Using the Validation Controls, Rich Controls, The Calendar, The AdRotator, Pages with Multiple Views, User Controls and Graphics, User Controls, Dynamic Graphics, The Chart Control, Website Navigation: Site Maps, URL Mapping and Routing, The SiteMapPath Control, The TreeView Control, The Menu Control.	**12**
III	**Error Handling, Logging, and Tracing** : Avoiding Common Errors, Understanding Exception Handling, Handling Exceptions, Throwing Your Own Exceptions, Using Page Tracing **State Management** : Understanding the Problem of State, Using View State, Transferring Information Between Pages, Using Cookies, Managing Session State, Configuring Session State, Using Application State, Comparing State Management Options **Styles, Themes, and Master Pages** : Styles, Themes, Master Page Basics, Advanced Master Pages	**12**
IV	**ADO.NET Fundamentals**: Understanding Databases, Configuring Your Database, Understanding SQL Basics, Understanding the Data Provider Model, Using Direct Data Access, Using Disconnected Data Access. **Data Binding**: Introducing Data Binding, Using Single-Value Data Binding, Using Repeated-Value Data Binding, Working with Data Source Controls, **The Data Controls**: The GridView, Formatting the GridView, Selecting a GridView Row, Editing with the GridView, Sorting and Paging the GridView, Using GridView Templates, The DetailsView and FormView	**12**
V	**XML**: XML Explained, The XML Classes, XML Validation, XML Display and Transforms. **Security Fundamentals**: Understanding Security Requirements, Authentication and Authorization, Forms Authentication, Windows Authentication. **ASP.NET AJAX**: Understanding Ajax, Using Partial Refreshes, Using Progress	**12**

	Notification, Implementing Timed Refreshes, Working with the ASP.NET AJAX Control Toolkit.	

Course Code	**Advanced Web Programming Practical List**
UIT5WPP	**1.Working with basic C# and ASP .NET** a. Create an application that obtains four int values from the user and displays the product. b. Create an application to demonstrate string operations. c. Create an application that receives the (Student Id, Student Name, Course Name, Date of Birth) information from a set of students. The application should also display the information of all the students once the data is entered. Create an application to demonstrate following operations i. Generate Fibonacci series. ii. Test for prime numbers. iii. Test for vowels. iv. Use of foreach loop with arrays v. Reverse a number and find sum of digits of a number. **2. Working with Object Oriented C# and ASP .NET** a. Create simple application to perform following operations i. Finding factorial Value ii. Money Conversion iii. Quadratic Equation iv. Temperature Conversion b. Create simple application to demonstrate use of following concepts i. Function Overloading ii. Inheritance (all types) iii. Constructor overloading iv. Interfaces c. Create simple application to demonstrate use of following concepts i. Using Delegates and events ii. Exception handling **3. Working with Web Forms and Controls** a. Create a simple web page with various sever controls to demonstrate setting and use of their properties. (Example :AutoPostBack) b. Demonstrate the use of Calendar control to perform following operations. a) Display messages in a calendar control b) Display vacation in a calendar control c) Selected day in a calendar control using style d) Difference between two calendar dates c. Demonstrate the use of Treeviewcontrol perform following operations. a) Treeview control and datalist b) Treeview operations **4. Working with Form Controls** a. Create a Registration form to demonstrate use of various Validation controls. b. Create Web Form to demonstrate use of Adrotator Control. c. Create Web Form to demonstrate use User Controls. **5. Working with Navigation, Beautification and Master page.** a. Create Web Form to demonstrate use of Website Navigation controls and Site Map. b. Create a web application to demonstrate use of Master Page with applying Styles and Themes for page beautification. c. Create a web application to demonstrate various states of ASP.NET Pages. **6. Working with Database** a. Create a web application bind data in a multiline textbox by querying in another textbox. b. Create a web application to display records by using database. c. Demonstrate the use of Datalist link control.

	7. Working with Database a. Create a web application to display Databinding using dropdownlist control. b. Create a web application for to display the phone no of an author using d database. c. Create a web application for inserting and deleting record from a database. (Using Execute-Non Query). **8. Working with data controls** a. Create a web application to demonstrate various uses and properties of SqlDataSource. b. Create a web application to demonstrate data binding using DetailsView and FormView Control. c. Create a web application to display Using Disconnected Data Access and Databinding using GridView. **9. Working with GridView control** a. Create a web application to demonstrate use of GridView control template andGridView hyperlink. b. Create a web application to demonstrate use of GridView button column and GridView events. c. Create a web application to demonstrate GridView paging and Creating own table format using GridView. **10. Working with AJAX and XML** a .Create a web application to demonstrate reading and writing operation with XML. b.Create a web application to demonstrate Form Security and Windows Security with proper Authentication and Authorization properties. c.Create a web application to demonstrate use of various Ajax controls. **11.Programs to create and use DLL**

Reference Books:

1) Beginning ASP.NET 4.5 in C# ,mathevmacDonald,Apress
2) Murach's ASP.NET 4.6 Web Programming in C#2015, SPD,SixthEdition,Mary Delamater and Anne Bohem

Course Code UIT5LSA	**Linux System Administration**

Objectives: To make the learners familiar with linux operating system administration . Learners will be able to install and configure different types of servers such as mail server,Web server etc. They will also be able to create and handle user accounts and maintain its security.

Expected Learning Outcomes:

1) Learners will be able to acquire skills to manage system level processes and handle software

	management on linux platforms. 2) Learners will be able to handle user accounts and manage storage space on systems. 3) Learners will be able to configure firewall and provide security to data on linux machines through cryptography 4) Learners will be able to acquire skills to configure different types of servers. 5) Learners will be able to do shell level programming in Linux	
I	**Introduction to Red Hat Enterprise Linux:** Linux, Open Source and Red Hat, Origins of Linux, Distributions, Duties of Linux System Administrator. **Command Line:** Working with the Bash Shell, Getting the Best of Bash, Useful Bash Key Sequences, Working with Bash History, Performing Basic File System Management Tasks, Working with Directories, Piping and Redirection, Finding Files **System Administration Tasks:** Performing Job Management Tasks, System and Process Monitoring and Management, Managing Processes with ps, Sending Signals to Processes with the kill Command, Using top to Show Current System Activity, Managing Process Niceness, Scheduling Jobs, Mounting Devices, Working with Links, Creating Backups, Managing Printers, Setting Up System Logging, Setting Up Rsyslog, Common Log Files, Setting Up Logrotate **Managing Software:** Understanding RPM, Understanding Meta Package Handlers, Creating Your Own Repositories, Managing Repositories, Installing Software with Yum, Querying Software, Extracting Files from RPM Packages	**12**
II	**Configuring and Managing Storage:** Understanding Partitions and Logical Volumes, Creating Partitions, Creating File Systems, File Systems Overview, Creating File Systems, Changing File System Properties, Checking the File System Integrity, Mounting File Systems Automatically Through fstab, Working with Logical Volumes, Creating Logical Volumes, Resizing Logical Volumes, Working with Snapshots, Replacing Failing Storage Devices, Creating Swap Space, Working with Encrypted Volumes **Connecting to the Network:** Understanding NetworkManager, Working with Services and Runlevels, Configuring the Network with NetworkManager, Working with system-config-network, NetworkManager Configuration Files, Network Service Scripts, Networking from the Command Line, Troubleshooting Networking, Setting Up IPv6, Configuring SSH, Enabling the SSH Server, Using the SSH Client, Using PuTTY on Windows Machines, Configuring Key-Based SSH Authentication, Using Graphical Applications with SSH, Using SSH Port Forwarding, Configuring VNC Server Access **Working with Users, Groups, and Permissions:** Managing Users and Groups, Commands for User Management, Managing Passwords, Modifying and Deleting User Accounts, Configuration Files, Creating Groups, Using Graphical Tools for User, and Group Management, Using External Authentication Sources, the Authentication Process, sssd, nsswitch, Pluggable Authentication Modules, Managing Permissions, the Role of Ownership, Basic Permissions: Read, Write, and Execute, Advanced Permissions, Working with Access Control Lists, Setting Default Permissions with umask, Working with Attributes	**12**
III	**Securing Server with iptables:** Understanding Firewalls, Setting Up a Firewall with system-config-firewall, Allowing Services, Trusted Interfaces, Masquerading, Configuration Files, Setting Up a Firewall with iptables, Tables, Chains, and Rules, Composition of Rule, Configuration Example, Advanced iptables Configuration, Configuring Logging, The Limit Module, Configuring NAT **Setting Up Cryptographic Services:** Introducing SSL, Proof of Authenticity: the Certificate Authority, Managing Certificates with openssl, Creating a Signing Request,	**12**

	Working with GNU Privacy Guard, Creating GPG Keys, Key Transfer, Managing GPG Keys, Encrypting Files with GPG, GPG Signing, Signing RPM Files **Configuring Server for File Sharing:** What is NFS? Advantages and Disadvantages of NFS, Configuring NFS4, Setting Up NFSv4, Mounting an NFS Share, Making NFS Mounts Persistent, Configuring Automount, Configuring Samba, Setting Up a Samba File Server, Samba Advanced Authentication Options, Accessing Samba Shares, Offering FTP Services.	
IV	**Configuring DNS and DHCP:** Introduction to DNS, The DNS Hierarchy, DNS Server Types, The DNS Lookup Process, DNS Zone Types, Setting Up a DNS Server, Setting Up a Cache-Only Name Server, Setting Up a Primary Name Server, Setting Up a Secondary Name Server, Understanding DHCP, Setting Up a DHCP Server **Setting Up a Mail Server:** Using the Message Transfer Agent, the Mail Delivery Agent, the Mail User Agent, Setting Up Postfix as an SMTP Server, Working with Mutt, Basic Configuration, Internet Configuration, Configuring Dovecot for POP and IMAP **Configuring Apache on Red Hat Enterprise Linux:** Configuring the Apache Web Server, Creating a Basic Website, Understanding the Apache Configuration Files, Apache Log Files, Working with Virtual Hosts, Securing the Web Server with TLS Certificates, Configuring Authentication, Setting Up Authentication with .htpasswd, Configuring LDAP Authentication, Setting Up MySQL	**12**
V	**Introducing Bash Shell Scripting:** Introduction, Elements of a Good Shell Script, Executing the Script, Working with Variables and Input, Understanding Variables, Variables, Subshells, and Sourcing, Working with Script Arguments, Asking for Input, Using Command Substitution, Substitution Operators, Changing Variable Content with Pattern Matching, Performing Calculations, Using Control Structures, Using if...then...else, Using case, Using while, Using until, Using for, Configuring booting with GRUB. **High-Availability Clustering:** High-Availability Clustering, The Workings of High Availability, High-Availability Requirements, Red Hat High-Availability Add-on Software, Components, Configuring Cluster-Based Services, Setting Up Bonding, Setting Up Shared Storage, Installing the Red Hat High Availability Add-On, Building the Initial State of the Cluster, Configuring Additional Cluster Properties, Configuring a Quorum Disk, Setting Up Fencing, Creating Resources and Services, Troubleshooting a Nonoperational Cluster, Configuring GFS2 File Systems **Setting Up an Installation Server:** Configuring a Network Server as an Installation Server, Setting Up a TFTP and DHCP Server for PXE Boot, Installing the TFTP Server, Configuring DHCP for PXE Boot, Creating the TFTP PXE Server Content, Creating a Kickstart File, Using a Kickstart File to Perform an Automated, Installation, Modifying the Kickstart File with, system-config-kickstart, Making Manual Modifications to the Kickstart File	**12**

Course Code	**Linux System Administration Practical List**
UIT5LAP	**0 Installation of RHEL 6.X** **1 Graphical User Interface and Command Line Interface and Processes** a Exploring the Graphical Desktop b The Command Line Interface c Managing Processes **2 Storage Devices and Links, Backup and Repository** a Working with Storage Devices and Links

	b Making a Backup c Creating a Repository **3 Working with RPMsm Storage and Networking** a Using Query Options b Extracting Files From RPMs c Configuring and Managing Storage d Connecting to the Network **4 Working with Users, Groups, and Permissions** **5 Firewall and Cryptographic services** a Securing Server with iptables b Setting Up Cryptographic Services **6 Configuring Server for File Sharing** a Configuring NFS Server and Client b Configuring Samba c Configuring FTP **7 DNS, DHCP and Mail Server** a Configuring DNS b Configuring DHCP c Setting Up a Mail Server **8 Web Server** a Configuring Apache on Red Hat Enterprise Linux b Writing a Script to Monitor Activity on the Apache Web Server c Using the select Command **9 Shell Scripts and High-Availability Clustering** a Writing Shell Scripts b Configuring Booting with GRUB c Configuring High Availability Clustering **10 Setting Up an Installation Server** a Configuring Network Server as an Installation Server b Setting Up a TFTP and DHCP Server for PXE Boot

Reference Books:

1. Red Hat Enterprise Linux 6 Administration, Sander van Vugt,John Wiley and Sons
2. Red hat Linux Networking and System Administration," Terry Collings and Kurt Wall" ,3rd Edition,Wiley

Course Code UIT5ENJ	**Enterprise Java**

Objectives

The objective of this course is to acquire the knowledge of advanced java technologies used in developing and deploying enterprise commercial website.

Expected Learning Outcomes

1) Learners will be able to create servlet and develop java applications with database connectivity.
2) Learners will study the fundamentals and core concepts of cookies, session, file uploading, file downloading and request dispatcher

3) Learners will gain knowledge and experience required to develop and deploy JSP application using JSTL.
4) Learners will be able to develop and deploy EJB application with concepts of Interceptors, JNDI..
5) Learners will be able tomake students familiar with the development of application using concept of Persistence, Object/Relational Mapping, JPA and Hibernate.

I	**Understanding Java EE:** What is an Enterprise Application? What is java enterprise edition? Java EE Technologies, Java EE evolution, Glassfish server **Java EE Architecture, Server and Containers:** Types of System Architecture, Java EE Server, Java EE Containers. **Introduction to Java Servlets:** The Need for Dynamic Content, Java Servlet Technology, Why Servlets? What can Servlets do? **Servlet API and Lifecycle:** Java Servlet API, The Servlet Skeleton, The Servlet Life Cycle, A Simple Welcome Servlet **Working With Servlets:** Getting Started, Using Annotations Instead of Deployment Descriptor. **Working with Databases:** What Is JDBC? JDBC Architecture, Accessing Database, The Servlet GUI and Database Example.	**12**
II	**Request Dispatcher:** RequestDispatcher Interface, Methods of RequestDispatcher, RequestDispatcher Application. **COOKIES:** Kinds Of Cookies, Where Cookies Are Used? Creating Cookies Using Servlet, Dynamically Changing TheColors Of A Page **SESSION:** What Are Sessions? Lifecycle Of Http Session, Session Tracking With Servlet API, A Servlet Session Example **Working With Files:** Uploading Files, Creating an Upload File Application, Downloading Files, Creating a Download File Application. **Working With Non-Blocking I/O:** Creating a Non-Blocking Read Application, Creating The Web Application, Creating Java Class, Creating Servlets, Retrieving The File, Creating index.jsp	**12**
III	**Introduction To Java Server Pages:** Why use Java Server Pages? Disadvantages Of JSP, JSP v\s Servlets, Life Cycle of a JSP Page, How does a JSP function? How does JSP execute? About Java Server Pages **Getting Started With Java Server Pages:** Comments, JSP Document, JSP Elements, JSP GUI Example. **Action Elements:** Including other Files, Forwarding JSP Page to Another Page, Passing Parameters for other Actions, Loading a Javabean. **Implicit Objects, Scope And El Expressions:** Implicit Objects, Character Quoting Conventions, Unified Expression Language [Unified El], Expression Language. **Java Server Pages Standard Tag Libraries:** What is wrong in using JSP ScriptletTags? How JSTL Fixes JSP Scriptlet's Shortcomings? Disadvantages Of JSTL, Tag Libraries.	**12**
IV	**Introduction To Enterprise Javabeans:** Enterprise Bean Architecture, Benefits of Enterprise Bean, Types of Enterprise Bean, Accessing Enterprise Beans, Enterprise Bean Application, Packaging Enterprise Beans **Working With Session Beans:** When to use Session Beans? Types of Session Beans, Remote and Local Interfaces, Accessing Interfaces, Lifecycle of Enterprise Beans, Packaging Enterprise Beans, Example of Stateful Session Bean, Example of Stateless Session Bean, Example of Singleton Session Beans. **Working with Message Driven Beans:** Lifecycle of a Message Driven Bean, Uses of Message Driven Beans, The Message Driven Beans Example.	**12**

	Interceptors: Request And Interceptor, Defining An Interceptor, AroundInvoke Method, Applying Interceptor, Adding An Interceptor To An Enterprise Bean, Build and Run the Web Application. **Java Naming and Directory Interface**: What is Naming Service? What is Directory Service? What is Java Naming and Directory interface? Basic Lookup, JNDI Namespace in Java EE, Resources and JNDI, Datasource Resource Definition in Java EE.	
V	**Persistence, Object/Relational Mapping And JPA:** What is Persistence? Persistence in Java, Current Persistence Standards in Java, Why another Persistence Standards? Object/Relational Mapping, **Introduction to Java Persistence API:** The Java Persistence API, JPA, ORM, Database and the Application, Architecture of JPA, How JPA Works? JPA Specifications. **Writing JPA Application:** Application Requirement Specifications, Software Requirements, The Application Development Approach, Creating Database And Tables in Mysql, Creating a Web Application, Adding the Required Library Files, Creating a Javabean Class, Creating Persistence Unit [Persistence.Xml], Creating JSPS, The JPA Application Structure, Running The JPA Application. **Introduction to Hibernate:** What is Hibernate? Why Hibernate? Hibernate, Database and The Application, Components of Hibernate, Architecture of Hibernate, How Hibernate Works? **Writing Hibernate Application:** Application Requirement Specifications, Software Requirements, The Application Development Approach, Creating Database and Tables in Mysql, Creating a Web Application, Adding The Required Library Files, Creating a Javabean Class, Creating Hibernate Configuration File, Adding a Mapping Class, Creating JSPS, Running The Hibernate Application.	**12**

Course Code	**Enterprise Java Practical List**
UIT5EJP	**1. Implement the following Simple Servlet applications.** a. Create a simple calculator application using servlet. b. Create a servlet for a login page. If the username and password are correct then it says message “Hello <username>” else a message “login failed” c. Create a registration servlet in Java using JDBC. Accept the details such as Username, Password, Email, and Country from the user using HTML Form and store the registration details in the database. **2. Implement the following Servlet applications with Cookies and Sessions.** a. Using Request Dispatcher Interface create a Servlet which will validate the password entered by the user, if the user has entered "Servlet" as password, then he will be forwarded to Welcome Servlet else the user will stay on the index.html page and an error message will be displayed. b. Create a servlet that uses Cookies to store the number of times a user has visited servlet. c. Create a servlet demonstrating the use of session creation and destruction. Also check whether the user has visited this page first time or has visited earlier also using sessions. **3. Implement the Servlet IO and File applications.** a. Create a Servlet application to upload and download a file. b. Develop Simple Servlet Question Answer Application using Database. c. Create simple Servlet application to demonstrate Non-Blocking Read Operation. **4. Implement the following JSP applications.**

	a. Develop a simple JSP application to display values obtained from the use of intrinsic objects of various types. b. Develop a simple JSP application to pass values from one page to another with validations. (Name-txt, age-txt, hobbies-checkbox, email-txt, gender-radio button). c. Create a registration and login JSP application to register and authenticate the user based on username and password using JDBC. **5. Implement the following JSP JSTL and EL Applications.** a. Create an html page with fields, eno, name, age, desg, salary. Now on submit this data to a JSP page which will update the employee table of database with matching eno. b. Create a JSP page to demonstrate the use of Expression language. c. Create a JSP application to demonstrate the use of JSTL. **6. Implement the following EJB Applications.** a. Create a Currency Converter application using EJB. b. Develop a Simple Room Reservation System Application Using EJB. c. Develop simple shopping cart application using EJB [Stateful Session Bean]. **7. Implement the following EJB applications with different types of Beans.** a. Develop simple EJB application to demonstrate Servlet Hit count using Singleton Session Beans. b. Develop simple visitor Statistics application using Message Driven Bean [Stateless Session Bean]. c. Develop simple Marks Entry Application to demonstrate accessing Database using EJB. **8 Implement the following JPA applications.** a Develop a simple Inventory Application Using JPA. b Develop a Guestbook Application Using JPA. c Create simple JPA application to store and retrieve Book details. **9 Implement the following JPA applications with ORM and Hibernate.** a Develop a JPA Application to demonstrate use of ORM associations. b Develop a Hibernate application to store Feedback of Website Visitor in MySQL Database. c Develop a Hibernate application to store and retrieve employee details in MySQL Database. **10 Implement the following Hibernate applications.** a Develop an application to demonstrate Hibernate One- To -One Mapping Using Annotation. b Develop Hibernate application to enter and retrieve course details with ORM Mapping. c Develop a five page web application site using any two or three Java EE Technologies.

Reference Books:

1) "Java EE 7 For Beginners", Sharanam Shah, Vaishali Shah, FirstEdition,SPD
2) "Advanced Java Programming",Uttam Kumar Roy, Oxford Press

Semester- VI

Course Code USIT6SQA	**Software Quality Assurance**
Objectives: The course is designed to introduce concepts about quality as the driving force behind success of software product, also focuses on life cycle of testing and different testing methodologies used for various test processes. **Expected Learning Outcomes:** 1) Learners will be able to analyse the quality of software product 2) Learners will be able to understand different testing methodology 3) Learners will be able to analyse the difference between black box and white box testing	

4) Learners will be able to understand verification and validation techniques 5) Learners will be able to understand special types of testing and levels of testing		
I	**Introduction to Quality:** Historical Perspective of Quality, What is Quality? (Is it a fact or perception?), Definitions of Quality, Core Components of Quality, Quality View, Financial Aspect of Quality, Customers, Suppliers and Processes, Total Quality Management (TQM), Quality Principles of Total Quality Management, Quality Management Through Statistical Process Control, Quality Management Through Cultural Changes, Continual (Continuous) Improvement Cycle, Quality in Different Areas, Benchmarking and Metrics, Problem Solving Techniques, Problem Solving Software Tools. **Software Quality:** Introduction, Constraints of Software Product Quality Assessment, Customer is a King, Quality and Productivity Relationship, Requirements of a Product, Organisation Culture, Characteristics of Software, Software Development Process, Types of Products, Schemes of Criticality Definitions, Problematic Areas of Software Development Life Cycle, Software Quality Management, Why Software Has Defects? Processes Related to Software Quality, Quality Management System Structure, Pillars of Quality Management System, Important Aspects of Quality Management.	**12**
II	**Fundamentals of testing:** Introduction, Necessity of testing, What is testing? Fundamental test process, The psychology of testing, Historical Perspective of Testing, Definitions of Testing, Approaches to Testing, Testing During Development Life Cycle, Requirement Traceability Matrix, Essentials of Software Testing, Workbench, Important Features of Testing Process, Misconceptions About Testing, Principles of Software Testing, Salient Features of Good Testing, Test Policy, Test Strategy or Test Approach, Test Planning, Testing Process and Number of Defects Found in Testing, Test Team Efficiency, Mutation Testing, Challenges in Testing, Test Team Approach, Process Problems Faced by Testing, Cost Aspect of Testing, Establishing Testing Policy, Methods, Structured Approach to Testing, Categories of Defect, Defect, Error, or Mistake in Software, Developing Test Strategy, Developing Testing Methodologies (Test Plan), Testing Process, Attitude Towards Testing (Common People Issues), Test Methodologies/Approaches, People Challenges in Software Testing, Raising Management Awareness for Testing, Skills Required by Tester, Testing throughout the software life cycle, Software development models, Test levels, Test types, the targets of testing, Maintenance testing	**12**
III	**Unit Testing: Boundary Value Testing:** Normal Boundary Value Testing, Robust Boundary Value Testing, Worst-Case Boundary Value Testing, Special Value Testing, Examples, Random Testing, Guidelines for Boundary Value Testing, **Equivalence Class Testing:** Equivalence Classes, Traditional Equivalence Class Testing, Improved Equivalence Class Testing, Edge Testing, Guidelines and Observations. **Decision Table–Based Testing:** Decision Tables, Decision Table Techniques, Cause-and-Effect Graphing, Guidelines and Observations, **Path Testing:** Program Graphs, DD-Paths, Test Coverage Metrics, Basis Path Testing, Guidelines and Observations, **Data Flow Testing:** Define/Use Testing, Slice-Based Testing, Program Slicing Tools.	**12**
IV	**Software Verification and Validation:** Introduction, Verification, Verification Workbench, Methods of Verification, Types of reviews on the basis od Stage Phase, Entities involved in verification, Reviews in testing lifecycle, Coverage in	**12**

	Verification, Concerns of Verification, Validation, Validation Workbench, Levels of Validation, Coverage in Validation, Acceptance Testing, Management of Verification and Validation, Software development verification and validation activities. **V-test Model:** Introduction, V-model for software, Testing during Proposal stage, Testing during requirement stage, Testing during test planning phase, Testing during design phase, Testing during coding, VV Model, Critical Roles and Responsibilities.	
V	**Levels of Testing:** Introduction, Proposal Testing, Requirement Testing, Design Testing, Code Review, Unit Testing, Module Testing, Integration Testing, Big-Bang Testing, Sandwich Testing, Critical Path First, Sub System Testing, System Testing, Testing Stages. **Special Tests:** Introduction, GUI testing, Compatibility Testing, Security Testing, Performance Testing, Volume Testing, Stress Testing, Recovery Testing, Installation Testing, Requirement Testing, Regression Testing, Error Handling Testing, Manual Support Testing, Intersystem Testing, Control Testing, Smoke Testing, Adhoc Testing, Parallel Testing, Execution Testing, Operations Testing, Compliance Testing, Usability Testing, Decision Table Testing, Documentation Testing, Training testing, Rapid Testing, Control flow graph, Generating tests on the basis of Combinatorial Designs, State Graph, Risk Associated with New Technologies, Process maturity level of Technology, Testing Adequacy of Control in New technology usage, Object Oriented Application Testing, Testing of Internal Controls, COTS Testing, Client Server Testing, Web Application Testing, Mobile Application Testing, eBusinesseCommerce Testing, Agile Development Testing, Data Warehousing Testing.	**12**

Course Code	**Project Implementation**
UIT6PIP	Chapter 1 to 4 should be submitted in Semester V in spiral binding. These chapter have also to be included in Semester VI report. Semester VI report has to be hard bound with golden embossing. Students will be evaluated based on the dissertation in semester V and dissertation and viva voce in Semester VI. **I. OBJECTIVES** **II. Type of the Project** **III. SOFTWARE AND BROAD AREAS OF APPLICATION FRONT END** **IV. Introduction** **1.1 PROJECT REPORT:** **CHAPTER 1: INTRODUCTION** **CHAPTER 2: SURVEY OF TECHNOLOGIES** **CHAPTER 3: REQUIREMENTS AND ANALYSIS** **CHAPTER 4: SYSTEM DESIGN** **CHAPTER 5: IMPLEMENTATION AND TESTING** **CHAPTER 6: RESULTS AND DISCUSSION** **CHAPTER 7: CONCLUSIONS**

	V. EXPLANATION OF CONTENTS **The details are given in Appendix–I**

Reference Books:

1)Software Testing: Principles, Techniques and Tools M. G. Limaye TMH 2017
2) Software Testing and Continuous Quality Improvement William E. Lewis Third Edition 2016
3) Software Testing: A Craftsman's Approach Paul C. Jorgenson 4th Edition 2017

Course Code USIT6SIC	**Security in Computing**	
Objectives: The course is designed to introduce the fundamentals about the security in computing for the network, hardware,applications,devices& data with the help of different types of security. **Expected Learning Outcomes:** 1) Learners will be able to introduce to basics of information security with risk analysis and design 2) Learners will be able to identify some of the factors driving the need for Database and storage security 3) Learners will be able to identify some of the factors driving the need for Network security 4) Learners will be able to gather information about multiple attacks, vulnerabilities and how to detect & prevent them. 5) Learners will be aware of information about cloud storage, virtualization and how to secure them		
I	**Information Security Overview** : The Importance of Information Protection, The Evolution of Information Security, Justifying Security Investment, Security Methodology, How to Build a Security Program, The Impossible Job, The Weakest Link, Strategy and Tactics, Business Processes vs. Technical Controls. **Risk Analysis**: Threat Definition, Types of Attacks, Risk Analysis. **Secure Design Principles**: The CIA Triad and Other Models, Defense Models, Zones of Trust, Best Practices for Network Defense.	**12**
II	**Authentication and Authorization**: Authentication, Authorization **Encryption**: A Brief History of Encryption, Symmetric-Key Cryptography, Public Key Cryptography, Public Key Infrastructure. **Storage Security**: Storage Security Evolution, Modern Storage Security, Risk Remediation, Best Practices. **Database Security**: General Database Security Concepts, Understanding Database Security Layers, Understanding Database-Level Security, Using Application Security, Database Backup and Recovery, Keeping Your Servers Up to Date, Database Auditing and Monitoring.	**12**
III	**Secure Network Design**: Introduction to Secure Network Design, Performance, Availability, Security. **Network Device Security**: Switch and Router Basics, Network Hardening. **Firewalls**: Overview, The Evolution of Firewalls, Core Firewall Functions, Additional Firewall Capabilities, Firewall Design. **Wireless Network Security**: Radio Frequency Security Basics, Data-Link Layer Wireless Security Features, Flaws, and Threats, Wireless Vulnerabilities and Mitigations, Wireless Network Hardening Practices and Recommendations, Wireless Intrusion Detection and Prevention, Wireless Network Positioning and Secure Gateways.	**12**
IV	**Intrusion Detection and Prevention Systems**: IDS Concepts, IDS Types and Detection Models, IDS Features, IDS Deployment Considerations, Security Information and Event Management (SIEM). **Voice over IP (VoIP) and PBX Security**: Background, VoIP Components, VoIP	**12**

	Vulnerabilities and Countermeasures, PBX, TEM: Telecom Expense Management. **Operating System Security Models**: Operating System Models, Classic Security Models, Reference Monitor, Trustworthy Computing, International Standards for Operating System Security.	
V	**Virtual Machines and Cloud Computing**: Virtual Machines, Cloud Computing. **Secure Application Design**: Secure Development Lifecycle, Application Security Practices, Web Application Security, Client Application Security, Remote Administration Security. **Physical Security**: Classification of Assets, Physical Vulnerability Assessment, Choosing Site Location for Security, Securing Assets: Locks and Entry Controls, Physical Intrusion Detection.	**12**

Course Code	**Security in ComputingPractical List**
UIT6SCP	**1 Configure Routers** a OSPF MD5 authentication. b NTP. c To log messages to the syslog server. d To support SSH connections. **2 Configure AAA Authentication** a Configure a local user account on Router and configure authenticate on the console and vty lines using local AAA b Verify local AAA authentication from the Router console and the PC-A client **3 Configuring Extended ACLs** a Configure, Apply and Verify an Extended Numbered ACL **4 Configure IP ACLs to Mitigate Attacks and IPV6 ACLs** a Verify connectivity among devices before firewall configuration. b Use ACLs to ensure remote access to the routers is available only from management station PC-C. c Configure ACLs on to mitigate attacks. d Configuring IPv6 ACLs **5 Configuring a Zone-Based Policy Firewall** **6 Configure IOS Intrusion Prevention System (IPS) Using the CLI** a Enable IOS IPS. b Modify an IPS signature. **7 Layer 2 Security** a Assign the Central switch as the root bridge. b Secure spanning-tree parameters to prevent STP manipulation attacks. c Enable port security to prevent CAM table overflow attacks. **8 Layer 2 VLAN Security** **9 Configure and Verify a Site-to-Site IPsec VPN Using CLI** **10 Configuring ASA Basic Settings and Firewall Using CLI** a Configure basic ASA settings and interface security levels using CLI b Configure routing, address translation, and inspection policy using CLI c Configure DHCP, AAA, and SSH d Configure a DMZ, Static NAT, and ACLs

Reference Books:

1. The Complete Reference: Information Security ,Mark Rhodes-Ousley, 2nd,McGraw-Hill

2. Essential Cybersecurity Science, Josiah Dykstra , Fifth, O'Reilly

3. Principles of Computer Security: CompTIA Security+ and Beyond ,Wm.ArthurConklin, Greg White, Second,McGraw Hill

Course Code UIT6BUI	**Business Intelligence**	
Objectives: To provide graduate students of M.Sc. Information Systems with comprehensive and in-depth knowledge of Business Intelligence (BI) principles and techniques by introducing the relationship between managerial and technological perspectives. This course is also designed to expose students to the frontiers of BI-intensive BIG data computing and information systems, while providing a sufficiently strong foundation to encourage further research. **Expected Learning Outcomes:** 1) Learners will be able to Identify the major frameworks of computerized decision support: decision support systems (DSS), data analytics and business intelligence. 2) Learners will be able to analyze data, choose relevant models and algorithms for respective applications. 3) Learners will be able to become familiar with classification methods, clustering methods. 4) Learners will be able to design application using Business Intelligence techniques. 5) Learners will be able to ability to design and develop the AI applications in real world scenario		
I	**Business intelligence:** Effective and timely decisions, Data, information and knowledge, The role of mathematical models, Business intelligence architectures, Ethics and business intelligence **Decision support systems:** Definition of system, Representation of the decision-making process, Evolution of information systems, Definition of decision support system, Development of a decision support system	**12**
II	**Mathematical models for decision making:** Structure of mathematical models, Development of a model, Classes of models **Data mining:** Definition of data mining, Representation of input data , Data mining process, Analysis methodologies **Data preparation**: Data validation, Data transformation, Data reduction	**12**
III	**Classification**: Classification problems, Evaluation of classification models, Bayesian methods, Logistic regression, Neural networks, Support vector machines **Clustering:** Clustering methods, Partition methods, Hierarchical methods, Evaluation of clustering models	**12**
IV	**Business intelligence applications:** **Marketing models**: Relational marketing, Sales force management, **Logistic and production models**: Supply chain optimization, Optimization models for logistics planning, Revenue management systems. **Data envelopment analysis**: Efficiency measures, Efficient frontier, The CCR model, Identification of good operating practices	**12**
V	**Knowledge Management:** Introduction to Knowledge Management, Organizational Learning and Transformation, Knowledge Management Activities, Approaches to Knowledge Management, Information Technology (IT) In Knowledge Management, Knowledge Management Systems Implementation, Roles of People in Knowledge Management **Artificial Intelligence and Expert Systems:** Concepts and Definitions of Artificial Intelligence, Artificial Intelligence Versus Natural Intelligence, Basic Concepts of Expert Systems, Applications of Expert	**12**

	Systems, Structure of Expert Systems, Knowledge Engineering, Development of Expert Systems	

Course Code	**Business Intelligence Practical List**
UIT6BIP	1)Import the legacy data from different sources such as (Excel , SqlServer, Oracle etc.) and load in the target system. (You can download sample database such as Adventureworks, Northwind, foodmart etc.) 2) Perform the Extraction Transformation and Loading (ETL) process to construct the database in the Sqlserver. 3)a. Create the Data staging area for the selected database. b. Create the cube with suitable dimension and fact tables based on ROLAP, MOLAP and HOLAP model. 4) a.Create the ETL map and setup the schedule for execution. b. Execute the MDX queries to extract the data from the datawarehouse. 5)a. Import the datawarehouse data in Microsoft Excel and create the Pivot table and Pivot Chart. b. Import the cube in Microsoft Excel and create the Pivot table and Pivot Chart to perform data analysis. 6) Apply the what – if Analysis for data visualization. Design and generate necessary reports based on the data warehouse data. 7)Perform the data classification using classification algorithm. 8)Perform the data clustering using clustering algorithm. 9)Perform the Linear regression on the given data warehouse data. 10)Perform the logistic regression on the given data warehouse data.

Reference Books:

1)Business Intelligence: Data Mining and Optimization for Decision Making, Carlo Vercellis, First Edition , Wiley
2) Decision support and Business Intelligence Systems, Efraim Turban, Ramesh Sharda, DursunDelen , Ninth, Pearson

Course Code USIT6GIS	**Principles of Geographic Information Systems**	
Objectives: The course is designed to introduce concepts about geographic information system to explore mapping data, analyze different mapping information and to create meaningful maps using different data. **Expected Learning Outcomes:** 1) Learners will be able to get introduction about basic GIS data types and technologies 2) Learners will be able to get knowledge about various GIS data management and processing techniques. 3) Learners will be able to Learn Spatial data processing techniques and positioning. 4) Learners will be able to learn various functions in GIS. 5) Learners will be able to create various maps in GIS		
I	**A Gentle Introduction to GIS** **The nature of GIS:** Some fundamental observations, Defining GIS, GISystems, GIScience and GIApplications, Spatial data and Geoinformation. **The real world and representations of it:** Models and modelling, Maps, Databases, Spatial databases and spatial analysis **Geographic Information and Spatial Database** **Models and Representations of the real world** **Geographic Phenomena:** Defining geographic phenomena, types of geographic phenomena, Geographic fields, Geographic objects, Boundaries **Computer Representations of Geographic Information:** Regular tessellations, irregular tessellations, Vector representations, Topology and Spatial relationships, Scale and Resolution, Representation of Geographic fields, Representation of Geographic objects **Organizing and Managing Spatial Data** **The Temporal Dimension**	**12**
II	**Data Management and Processing Systems** **Hardware and Software Trends** **Geographic Information Systems:** GIS Software, GIS Architecture and functionality, Spatial Data Infrastructure (SDI) **Stages of Spatial Data handling:** Spatial data handling and preparation, Spatial Data Storage and maintenance, Spatial Query and Analysis, Spatial Data Presentation. **Database management Systems:** Reasons for using a DBMS, Alternatives for data management, The relational data model, Querying the relational database. **GIS and Spatial Databases:** Linking GIS and DBMS, Spatial database functionality.	**12**
III	**Spatial Referencing and Positioning** **Spatial Referencing:** Reference surfaces for mapping, Coordinate Systems, Map Projections, Coordinate Transformations **Satellite-based Positioning:** Absolute positioning, Errors in absolute positioning, Relative positioning, Network positioning, code versus phase measurements, Positioning technology **Data Entry and Preparation**	**12**

	Spatial Data Input: Direct spatial data capture, Indirect spatial data captiure, Obtaining spatial data elsewhere **Data Quality:** Accuracy and Positioning, Positional accuracy, Attribute accuracy, Temporal accuracy, Lineage, Completeness, Logical consistency **Data Preparation:** Data checks and repairs, Combining data from multiple sources **Point Data Transformation:** Interpolating discrete data, Interpolating continuous data	
IV	**Spatial Data Analysis** **Classification of analytical GIS Capabilities** **Retrieval, classification and measurement:** Measurement, Spatial selection queries, Classification **Overlay functions:** Vector overlay operators,Raster overlay operators **Neighbourhood functions:** Proximity computations, Computation of diffusion, Flow computation, Raster based surface analysis **Analysis:** Network analysis, interpolation, terrain modeling **GIS and Application models:** GPS, Open GIS Standards, GIS Applications and Advances **Error Propagation in spatial data processing:** How Errors propagate, Quantifying error propagation	**12**
V	**Data Visualization** **GIS and Maps, The Visualization Process** **Visualization Strategies: Present or explore?** **The cartographic toolbox:** What kind of data do I have?, How can I map my data? **How to map?:**How to map qualitative data, How to map quantitative data, How to map the terrain elevation, How to map time series **Map Cosmetics, Map Dissemination**	**12**

Course Code	**Principles Geographic Information SystemsPractical List**
USIT6GIP	**1.**Creating and Managing Vector Data: Adding vector layers, setting properties, formatting, calculating line lengths and statistics **2.** Exploring and Managing Raster data: Adding raster layers, raster styling and analysis, raster mosaicking and clipping method. **3.**Making a Map, Working with Attributes, Importing Spreadsheets or CSV files Using Plugins, Searching and Downloading OpenStreetMap Data **4.** Working with attributes, terrain Data **5.** Working with Projections and WMS Data **6**GeoreferencingTopo Sheets and Scanned Maps ,Georeferencing Aerial Imagery, Digitizing Map Data . **7.** Managing Data Tables and Spatial data Sets: Table joins, spatial joins, points in Polygon analysis, performing spatial queries. **8.** Advanced GIS Operations 1: Nearest Neighbour Analysis, Sampling Raster Data usingPoints or Polygons, Interpolating Point Data. **9.** Advance GIS Operations 2: Batch Processing using Processing Framework Automating,Complex Workflows using Processing Modeler**,**Automating Map Creation with Print Composer Atlas. **10.** Validating Map data

Reference Books:

1) Principles of Geographic Information Systems, Otto Huisman and Rolf A. , The International Institute of Geoinformation Science and Earth Observation, Fourth edition,2009.
2) Principles of Geographic Information Systems , P.A Burrough and R.A.McDonnell , Oxford University Press , Third edition,1999.
3) Fundamentals of Spatial Information Systems, R.Laurini and D. Thompson, Academic Press,1994.
4) Fundamentals of Geographic Information Systems, Michael N.Demers , Wiley Publications , Fourth edition , 2009 .
5) Introduction to Geographic Information Systems , Chang Kang-tsung (Karl), McGrawHill edition, Any above 3rd Edition.
6) GIS Fundamentals: A First Text on Geographic Information Systems , Paul Bolsatd , XanEdu Publishing Inc , 5th Edition .

CourseCode UIT6ISM	**IT Service Management**	
Objectives The objective of this course is to introduce the concept of the IT Service Management with respect to the services, to improve the end user experience and connecting IT to wider business objectives. **Expected Learning Outcomes** 1) Learners will be able to gain understanding of scope, purpose, and objective of Service Management. 2) Learners will be able to understand Service Design, Service Design Principles and its Strategies. 3) Learners will be able to understand implementation of services through Service Transition Phase. 4) Learners will be able to understand Service Operation Phase and activities for operating services. 5) Learners will be able to understand Process of Continual Service Improvement and its challenges delivering the service.		
I	**IT Service Management:** Introduction, What is service management? What are services? Business Process, Principles of Service management: Specialisation and Coordination, The agency principle, Encapsulation, Principles of systems, The service Life Cycle, Functions and processes across the life cycle. **Service Strategy Principles:** Value creation, Service Assets, Service Provider Service Structures, Service Strategy Principles. **Service Strategy:** Define the market, Develop the offerings, Develop Strategic Assets, Prepare for execution. **Challenges, Critical Success factors and risks:** Complexity, Coordination and Control, Preserving value, Effectiveness in measurement, Risks.	**12**
II	**Service Design:** Fundamentals, **Service Design Principles:** Goals, Balanced Design, Identifying Service requirements, identifying and documenting business requirements and drivers, Design activities, Design aspects, Subsequent design activities, Design constraints, Service oriented architecture, Business Service Management, Service Design Models **Service Design Processes:** Service Catalogue Management, Service Level Management, Capacity Management, Availability Management, IT Service Continuity Management, Information Security Management, Supplier Management **Challenges, Critical Success factors and risks:** Challenges, Risks	**12**
II	**Service Transition:** Fundamentals, **Service Transition Principles:** Principles Supporting Service Transition, Policies for Service Transition **Service Transition Processes:** Transition planning and support, Change Management, Service Asses Configuration Management, Service and Deployment Management, Service Validation and Testing, Evaluation, Knowledge Management. **Challenges, Critical Success factors and risks:** Challenges, Critical Success factors, Risks, Service Transition under difficult Conditions.	**12**
IV	**Service Operation:** Fundamentals, **Service Operation Principles:** Functions, groups, teams, departments and divisions, Achieving balance in service operations, Providing service, Operation staff involvement in service design and service transition, Operational	**12**

	Health, Communication, Documentation **Service Operation Processes:** Event Management, Incident Management, Request fulfilment, Problem Management, Access Management, Operational activities of processes covered in other lifecycle phases. **Challenges, Critical Success factors and risks:** Challenges, Critical Success factors, Risks	
V	**Continual Service Improvement(CSI) Principles:** CSI Approach, CSI and organizational change, Ownership, CSI register, External and Internal drivers, Service level management, Knowledge management, The Deming cycle, Service Measurement, IT governance, Frameworks, models, standards and quality Systems, CSI inputs and outputs. **CSI Process:** The seven step improvement process. **CSI Methods nad Techniques:** Methods and techniques, Assessments, benchmarking, Service Measurement, Metrics, Return on Investment, Service reporting, CSI and other service management processes, **Organising for CSI:** Organisational development, Functions, roles, Customer Engagement, Responsibility model - RACI, Competence and training. **Technology considerations:** Tools to support CSI activities. **Implementing CSI:** Critical Considerations for implementing CSI, The start, Governance, CSI and organisational change, Communication Strategy and Plan	**12**

Course Code	**Advanced Mobile ProgrammingPractical List**
UIT6ISP	**1. Introduction to Android, Introduction to Android Studio IDE, Application Fundamentals**: Creating a Project, Android Components, Activities, Services, Content Providers, Broadcast Receivers, Interface overview, Creating Android Virtual device, USB debugging mode, Android Application Overview. Simple "Hello World" program. **2. Programming Resources** Android Resources: (Color, Theme, String, Drawable, Dimension, Image **3. Programming Activities and fragments** Activity Life Cycle, Activity methods, Multiple Activities, Life Cycle of fragments and multiple fragments. **4. Programs related to different Layouts** Coordinate, Linear, Relative, Table, Absolute, Frame, List View, Grid View. **5. Programming UI elements** AppBar, Fragments, UI Components **6. Programming menus, dialog, dialog fragments** **7. Programs on Intents, Events, Listeners and Adapters** The Android Intent Class, Using Events and Event Listeners **8. Programs on Services, notification and broadcast receivers** **9. Database Programming with SQLite** **10. Programming threads, handles and asynchronized programs** **11. Programming Media API and Telephone API** **12. Programming Security and permissions** **13. Programming Network Communications and Services (JSON)**

Reference Books:

1. "ITIL v3 Foundation Complete Certification Kit", 2009

2. "ITIL v3 Service Strategy", OGC/TSO
3. "ITIL v3 Service Transition", OGC/TSO
4. "ITIL v3 Service Operation", OGC/TSO
5. "ITIL Continual Service Improvement", TSO, 2011

APPENDIX –1

Project Dissertation Semester V and Project Implementation Semester VI

Chapter 1 to 4 should be submitted in Semester Vin spiral binding. These chapter have also to be included in Semester VI report. Semester VI report has to be hard bound with golden embossing. Students will be evaluated based on the dissertation in semester V and dissertation and viva voce in Semester VI.

I. OBJECTIVES

• Describe the Systems Development Life Cycle (SDLC).

• Evaluate systems requirements.

• Complete a problem definition.

• Evaluate a problem definition.

• Determine how to collect information to determine requirements.

• Perform and evaluate feasibility studies like cost-benefit analysis, technical feasibility, time feasibility and Operational feasibility for the project.

• Work on data collection methods for fact finding.

• Construct and evaluate data flow diagrams.

• Construct and evaluate data dictionaries.

• Evaluate methods of process description to include structured English, decision tables and decision trees.

• Evaluate alternative tools for the analysis process.

• Create and evaluate such alternative graphical tools as systems flow charts and state transition diagrams.

• Decide the S/W requirement specifications and H/W requirement specifications.

• Plan the systems design phase of the SDLC.

• Distinguish between logical and physical design requirements.

• Design and evaluate system outputs.

• Design and evaluate systems inputs.

• Design and evaluate validity checks for input data.

• Design and evaluate user interfaces for input.

• Design and evaluate file structures to include the use of indexes.

• Estimate storage requirements.

• Explain the various file update processes based on the standard file organizations.

• Decide various data structures.

• Construct and evaluate entity-relationship (ER) diagrams for RDBMS related projects.

• Perform normalization for the unnormalized tables for RDBMS related projects

• Decide the various processing systems to include distributed, client/server, online and others.

• Perform project cost estimates using various techniques.

• Schedule projects using both GANTT and PERT charts.

• Perform coding for the project.

• Documentation requirements and prepare and evaluate systems documentation.

• Perform various systems testing techniques/strategies to include the phases of testing.

• Systems implementation and its key problems.

• Generate various reports.

• Be able to prepare and evaluate a final report.

• Brief the maintenance procedures and the role of configuration management in operations.

• To decide the future scope and further enhancement of the system.

• Plan for several appendices to be placed in support with the project report documentation.

• Decide the various processing systems to include distributed, client/server, online and others.

• Perform project cost estimates using various techniques.

• Schedule projects using both GANTT and PERT charts.

• Perform coding for the project.

• Documentation requirements and prepare and evaluate systems documentation.

• Perform various systems testing techniques/strategies to include the phases of testing.

• Systems implementation and its key problems.

• Generate various reports.

• Be able to prepare and evaluate a final report.

• Brief the maintenance procedures and the role of configuration management in operations.

• To decide the future scope and further enhancement of the system.

• Plan for several appendices to be placed in support with the project report documentation.

• Work effectively as an individual or as a team member to produce correct, efficient, well-organized and documented programs in a reasonable time.

• Recognize problems that are amenable to computer solutions, and knowledge of the tool necessary for solving such problems.

• Develop of the ability to assess the implications of work performed.

• Get good exposure and command in one or more application areas and on the software

• Develop quality software using the software engineering principles

• Develop of the ability to communicate effectively.

II. Type of the Project

The majority of the students are expected to work on a real-life project preferably in some industry/ Research and Development Laboratories/Educational Institution/Software Company. Students are encouraged to work in the areas listed below. However, it is ***not mandatory*** for a student to work on a real-life project. The student can formulate a project problem with the help of her/his Guide and submit the project proposal of the same. **Approval of the project proposal is mandatory**. If approved, the student can commence working on it, and complete it. Use the latest versions of the software packages for the development of the project.

III. SOFTWARE AND BROAD AREAS OF APPLICATION

FRONT END / GUI Tools	.Net Technologies,Java
DBMS/BACK END	Oracle, SQL Plus, MY SQL, SQL Server,
LANGUAGES	C, C++, Java, VC++, C#, R,Python
SCRIPTING LANGUAGES	PHP,JSP, SHELL Scripts (Unix), TcL/TK,
.NET Platform	F#,C#. Net, Visual C#. Net, ASP.Net
MIDDLE WARE (COMPONENT) TECHNOLOGIES	COM/DCOM, Active-X, EJB
UNIX INTERNALS	Device Drivers, RPC, Threads, Socket
NETWORK/WIRELESS TECHNOLOGIES	-

IV. Introduction

The project report should be documented with scientific approach to the solution of the problem that the students have sought to address. The project report should be prepared in order to solve the problem in a methodical and professional manner, making due references to appropriate techniques, technologies and professional standards. The student should start the documentation process from the first phase of software development so that one can easily identify the issues to be focused upon in the ultimate project report. The student should also include the details from the project diary, in which they will record the progress of their project throughout the course. The project report should contain enough details to enable examiners to evaluate the work. The important points should be highlighted in the body of the report, with details often referred to appendices.

1.1 PROJECT REPORT:

Title Page

Original Copy of the Approved Proforma of the Project Proposal

Certificate of Authenticated work

Role and Responsibility Form

Abstract

Acknowledgement

Table of Contents

Table of Figures

CHAPTER 1: INTRODUCTION

1.1 Background

1.2 Objectives

1.3 Purpose, Scope, and Applicability
1.3.1 Purpose

1.3.2 Scope

1.3.3 Applicability

1.4 Achievements

1.5 Organisation of Report

CHAPTER 2: SURVEY OF TECHNOLOGIES

CHAPTER 3: REQUIREMENTS AND ANALYSIS

3.1 Problem Definition

3.2 Requirements Specification

3.3 Planning and Scheduling

3.4 Software and Hardware Requirements

3.5 Preliminary Product Description

3.6 Conceptual Models

CHAPTER 4: SYSTEM DESIGN

4.1 Basic Modules

4.2 Data Design

4.2.1 Schema Design

4.2.2 Data Integrity and Constraints

4.3 Procedural Design

4.3.1 Logic Diagrams

4.3.2 Data Structures

4.3.3 Algorithms Design

4.4 User interface design

4.5 Security Issues

4.6 Test Cases Design

The documentation should use tools like star UML, Visuo for windows, Rational Rose for design as part of Software Project Management Practical Course. The documentation should

be spiral bound for semester V and the entire documentation should be hard bound during semester VI.

CHAPTER 5: IMPLEMENTATION AND TESTING

5.1 Implementation Approaches

5.2 Coding Details and Code Efficiency 68

5.2.1 Code Efficiency

5.3 Testing Approach

5.3.1 Unit Testing

5.3.2 Integrated Testing

5.3.3 Beta Testing

5.4 Modifications and Improvements

5.5 Test Cases

CHAPTER 6: RESULTS AND DISCUSSION

6.1 Test Reports

6.2 User Documentation

CHAPTER 7: CONCLUSIONS

7.1 Conclusion

7.1.1 Significance of the System

7.2 Limitations of the System

7.3 Future Scope of the Project

REFERENCES

GLOSSARY

APPENDIX A

APPENDIX B

V. EXPLANATION OF CONTENTS

Title Page

Sample format of Title page is given in Appendix 1 of this block. Students should follow the given format.

Original Copy of the Approved Proforma of the Project Proposal

Sample Proforma of Project Proposal is given in Appendix 2 of this block. Students should follow the given format.

Certificate of Authenticated work

Sample format of Certificate of Authenticated work is given in Appendix 3 of this block. Students should follow the given format.

Role and Responsibility Form

Sample format for Role and Responsibility Form is given in Appendix 4 of this block.

Students should follow the given format.

Abstract

This should be one/two short paragraphs (100-150 words total), summarising the project work. It is important that this is not just a re-statement of the original project outline. A suggested flow is background, project aims and main achievements. From the abstract, a reader should be able to ascertain if the project is of interest to them and, it should present results of which they may wish to know more details.

Acknowledgements

This should express student's gratitude to those who have helped in the preparation of project. Table of Contents: The table of contents gives the readers a view of the detailed structure of the report. The students would need to provide section and subsection headings with associated pages. The formatting details of these sections and subsections are given below. Table of Figures: List of all Figures, Tables, Graphs, Charts etc. along with their page numbers in a table of figures.

Chapter 1: Introduction

The introduction has several parts as given below:

Background: A description of the background and context of the project and its relation to work already done in the area. Summarise existing work in the area concerned with the project work.

Objectives: Concise statement of the aims and objectives of the project. Define exactly what is going to be done in the project; the objectives should be about 30 /40 words. Purpose, Scope and Applicability: The description of Purpose, Scope, and Applicability are given below:

• Purpose: Description of the topic of the project that answers questions on why this project is being done. How the project could improve the system its significance and theoretical framework.

• Scope: A brief overview of the methodology, assumptions and limitations. The students should answer the question: What are the main issues being covered in the project? What are the main functions of the project?

• Applicability: The student should explain the direct and indirect applications of their work. Briefly discuss how this project will serve the computer world and people. Achievements: Explain what knowledge the student achieved after the completion of the work. What contributions has the project made to the chosen area? Goals achieved -describes the degree to which the findings support the original objectives laid out by the project. The goals may be partially or fully achieved, or exceeded. Organisation of Report: Summarising the remaining chapters of the project report, in effect, giving the reader an overview of what is to come in the project report.

Chapter 2: Survey of Technologies

In this chapter Survey of Technologies should demonstrate the students awareness and understanding of Available Technologies related to the topic of the project. The student should give the detail of all the related technologies that are necessary to complete the project. The should describe the technologies available in the chosen area and present a comparative study of all those Available Technologies. Explain why the student selected the one technology for the completion of the objectives of the project.

Chapter 3: Requirements and Analysis

Problem Definition: Define the problem on which the students are working in the project. Provide details of the overall problem and then divide the problem in to sub-problems. Define each sub-problem clearly

Requirements Specification: In this phase the student should define the requirements of the system, independent of how these requirements will be accomplished. The Requirements Specification describes the things in the system and the actions that can be done on these things. Identify the operation and problems of the existing system.

Planning and Scheduling: Planning and scheduling is a complicated part of software development. Planning, for our purposes, can be thought of as determining all the small tasks that must be carried out in order to accomplish the goal. Planning also takes into account, rules, known as constraints, which, control when certain tasks can or cannot happen. Scheduling can be thought of as determining whether adequate resources are available to carry out the plan. The student should show the Gantt chart and Program Evaluation Review Technique (PERT).

Software and Hardware Requirements: Define the details of all the software and hardware needed for the development and implementation of the project.

• Hardware Requirement: In this section, the equipment, graphics card, numeric co-processor, mouse, disk capacity, RAM capacity etc. necessary to run the software must be noted.

• Software Requirements: In this section, the operating system, the compiler, testing tools, linker, and the libraries etc. necessary to compile, link and install the software must be listed.

Preliminary Product Description: Identify the requirements and objectives of the new system. Define the functions and operation of the application/system the students are developing as project.

Conceptual Models: The student should understand the problem domain and produce a model of the system, which describes operations that can be performed on the system, and the allowable sequences of those operations. Conceptual Models could consist of complete Data Flow Diagrams, ER diagrams, Object-oriented diagrams, System Flowcharts etc.

Chapter 4: System Design

Describes desired features and operations in detail, including screen layouts, business rules, process diagrams, pseudo code and other documentation.

Basic Modules: The students should follow the divide and conquer theory, so divide the overall problem into more manageable parts and develop each part or module separately. When all modules are ready, the student should integrate all the modules into one system. In this phase, the student should briefly describe all the modules and the functionality of these modules. Data Design: Data design will consist of how data is organised, managed and manipulated.

• Schema Design: Define the structure and explanation of schemas used in the project.

• Data Integrity and Constraints: Define and explain all the validity checks and constraints provided to maintain data integrity.

Procedural Design: Procedural design is a systematic way for developing algorithms or procedurals.

• Logic Diagrams: Define the systematically flow of procedure that improves its comprehension and helps the programmer during implementation. e.g., Control Flow Chart, Process Diagrams etc.

• Data Structures: Create and define the data structure used in procedures.

• Algorithms Design: With proper explanations of input data, output data, logic of processes, design and explain the working of algorithms.

User Interface Design: Define user, task, environment analysis and how to map those requirements in order to develop a “User Interface”. Describe the external and internal components and the architecture of user interface. Show some rough pictorial views of the user interface and its components.

Security Issues: Discuss Real-time considerations and Security issues related to the project and explain how the student intends avoiding those security problems. What are the security policy plans and architecture?

Test Cases Design: Define test cases, which will provide easy detection of errors and mistakes with in a minimum period of time and with the least effort. Explain the different conditions in which the students wish to ensure the correct working of the project.

Chapter 5: Implementation and Testing

Implementation Approaches: Define the plan of implementation, and the standards the students have used in the implementation.

Coding Details and Code Efficiency: Students not need include full source code, instead, include only the important codes (algorithms, applets code, forms code etc). The program code should contain comments needed for explaining the work a piece of code does. Comments may be needed to explain why it does it, or, why it does a particular way.

The student can explain the function of the code with a shot of the output screen of that program code.

• Code Efficiency: The student should explain how the code is efficient and how the students have handled code optimisation.

Testing Approach: Testing should be according to the scheme presented in the system design chapter and should follow some suitable model – e.g., category partition, state machine-based. Both functional testing and user-acceptance testing are appropriate. Explain the approach of testing.

• Unit Testing: Unit testing deals with testing a unit or module as a whole. This would test the interaction of many functions but, do confine the test within one module.

• Integrated Testing: Brings all the modules together into a special testing environment, then checks for errors, bugs and interoperability. It deals with tests for the entire application. Application limits and features are tested here.

Modifications and Improvements: Once the students finish the testing they are bound to be faced with bugs, errors and they will need to modify your source code to improve the system. Define what modification are implemented in the system and how it improved the system.

Chapter 6: Results and Discussion

Test Reports: Explain the test results and reports based on the test cases, which should show that the project is capable of facing any problematic situation and that it works fine in different conditions. Take the different sample inputs and show the outputs.

User Documentation: Define the working of the software; explain its different functions, components with screen shots. The user document should provide all the details of the product in such a way that any user reading the manual, is able to understand the working and functionality of the document.

Chapter 7: Conclusions

Conclusion: The conclusions can be summarised in a fairly short chapter (2 or 3 pages). This chapter brings together many of the points that would have made in the other chapters. Limitations of the System: Explain the limitations encountered during the testing of the project that the students were not able to modify. List the criticisms accepted during the demonstrations of the project.

Future Scope of the Project describes two things: firstly, new areas of investigation prompted by developments in this project, and secondly, parts of the current work that was not completed due to time constraints and/or problems encountered.

PROFORMA FOR THE APPROVAL PROJECT PROPOSAL
(Note: All entries of the proforma of approval should be filled up with appropriate and complete information. Incomplete proforma of approval in any respect will be summarily rejected.)
PNR **No.: ……** Roll no**: _____**
1. Name of the Student

_____
2. Title of the Project

_____
3. Name of the Guide

_____
4. Teaching experience of the Guide _____

5. Is this your first submission? Yes No

Signature of the Student
Date: ……

Signature of the Guide
Date: …….

Signature of the Coordinator
Date: ……