SOPHIA COLLEGE FOR WOMEN (AUTONOMOUS)

Affiliated to

UNIVERSITY OF MUMBAI

Programme: Information Technology

Programme Code: SBTTEC

F.Y.B.Sc. I.T.
2018-19

(Choice Based Credit System with effect from the year 2018-19)

Programme Outline: FYBSc IT (SEMESTER I)

Course Code	Unit No.	Name of the Unit	Credits
SBTTEC101		**IMPERATIVE PROGRAMMING**	2
	1	Introduction and fundamentals	
	2	Operators, Expressions, Data Input and Output	
	3	Conditional Statements, Loops and Functions	
	4	Program structure, Preprocessor, Arrays	
	5	Pointers, Structures and Unions	
SBTTEC102		**DIGITAL ELECTRONICS**	2
	1	Number system, Binary Arithmetic	
	2	Boolean Algebra and Logic GatesMinterm, Maxterm and Karnaugh Maps	
	3	Combinational Logic Circuits, Arithmetic Circuits	
	4	Multiplexer, Demultiplexer, ALU,Encoder and Decoder Sequential Circuits: Flip-Flop	
	5	Counters, Shift Register	
SBTTEC103		**OPERATING SYSTEMS**	2
	1	Introduction, Processes and Threads	
	2	Memory Management, File Systems	
	3	Input-Output, Deadlocks	
	4	Virtualization and Cloud Multiple Processor Systems	
	5	Case Study on LINUX andANDROID, Case Study on Windows	
SBTTEC104		**DISCRETE MATHEMATICS**	2
	1	Introduction, Set Theory The Logic of Compound Statements	

	2	Quantified Statements, Elementary Number Theory and Methods of Proof	
	3	Sequences, Mathematical Induction,and Recursion Functions	
	4	Relations, Graphs and Trees	
	5	Counting and Probability	
SBTTEC105		**COMMUNICATION SKILLS**	2
	1	The Seven Cs of EffectiveCommunication Understanding Business Communication	
	2	Writing Business Messages andDocuments Developing Oral Communication Skills for Business	
	3	Developing Oral CommunicationSkills for Business Understanding Specific Communication Needs	
	4	Understanding Specific Communication Needs	
	5	Presentation Process	
SBTTECP101		IMPERATIVE PROGRAMMING PRACTICAL	2
SBTTECP102		DIGITAL ELECTRONICS PRACTICAL	2
SBTTECP103		OPERATING SYSTEMS PRACTICAL	2
SBTTECP104		DISCRETE MATHEMATICS PRACTICAL	2
SBTTECP105		COMMUNICATION SKILLS PRACTICAL	2
		Total Credits	**20**

Programme Outline: FYBSc IT (SEMESTER II)

Course Code	Unit No.	Name of the Unit	Credits
SBTTEC201		**OBJECT ORIENTED ROGRAMMING**	2
	1	Object Oriented Methodology, Principles of OOPS	
	2	Classes and Objects, Constructors and Destructors	
	3	Polymorphism, Virtual Functions	
	4	Program development using Inheritance, Exception Handling	
	5	Templates, Working with Files	
SBTTEC202		**MICROPROCESSOR ARCHITECTURE**	2
	1	Microprocessor, microcomputers, and Assembly Language, Microprocessor Architecture andMicrocomputer System 8085 Microprocessor Architecture and Memory Interface	
	2	Interfacing of I/O Devices Introduction to 8085 Assembly Language Programming Introduction to 8085 Instructions	
	3	Programming Techniques WithAdditional Instructions Counters and Time Delays Stacks and Sub-Routines	
	4	Code Conversion, BCD Arithmetic,and 16-Bit Data Operations Software Development System andAssemblers Interrupts	
	5	The Pentium and Pentium Pro microprocessors Core 2 and later	

		Microprocessors SUN SPARC Microprocessor	
SBTTEC203		**WEB PROGRAMMING**	2
	1	Internet and the World Wide Web HTML5	
	2	HTML5 Page layout and navigation HTML5 Tables, Forms and Media	
	3	Java Script, Core JavaScript Properties and Methods, Events and Event Handlers	
	4	PHP	
	5	Advanced PHP and MySQL	
SBTTEC204		**NUMERICAL AND STATISTICAL METHODS**	2
	1	Mathematical Modeling and Engineering Problem Solving Approximations and Round-Off ErrorsTruncation Errors and the Taylor Series	
	2	Solutions of Algebraic and Transcendental Equations Interpolation	
	3	Solution of simultaneous algebraic equations (linear) using iterative methods, Numerical differentiation andIntegration Numerical solution of 1st and 2^{nd} order differential equations	
	4	Least-Squares Regression Linear Programming	
	5	Random variables Distributions	
SBTTEC205		**GREEN COMPUTING**	2
	1	Overview and Issues Initiatives and Standards	
	2	Minimizing Power Usage Cooling	

	3	Changing the Way of Work Going Paperless	
	4	Recycling Hardware Considerations	
	5	Greening Your Information Systems Staying Green	
SBTTECP201	1	OBJECT ORIENTEDPROGRAMMING PRACTICAL	2
SBTTECP202	2	MICROPROCESSOR ARCHITECTURE PRACTICAL	2
SBTTECP203	3	WEB PROGRAMMING PRACTICAL	2
SBTTECP204	4	NUMERICAL AND STATISTICAL METHODS PRACTICAL	2
SBTTECP205	5	GREEN COMPUTING PRACTICAL	2
		Total Credits	**20**

Preamble:

Information Technology (IT) refers to the use, development, and management of computer systems, software, and networks to process, store, retrieve, and exchange information. It encompasses a broad range of technologies and practices aimed at solving problems, improving efficiency, and enabling communication within and between organizations and individuals.
In an era marked by rapid digital transformation and technological advancements, our program is designed to equip students with a comprehensive understanding of the foundational and emerging concepts in Information Technology.
Our BSc IT curriculum integrates theoretical knowledge with practical skills, preparing students to tackle real-world challenges and excel in a diverse range of IT careers. Through a combination of rigorous coursework, industry-relevant projects, and learning experiences, we aim to develop well-rounded professionals who are adept at problem-solving and equipped with the tools to drive technological innovation.

PROGRAMME OBJECTIVES

PO 1	To think analytically and creatively in developing robust, extensible and maintainable technological solutions to simple and complex problems.
PO 2	To work effectively as a part of a team to achieve a common stated goal.
PO 3	To imbibe quality software development practices.
PO 4	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.
PO 5	To communicate effectively with a range of audiences both technical and non-technical.

PROGRAMME SPECIFIC OUTCOMES

PSO 1	The Learner will be able to demonstrate a strong understanding of fundamental concepts in information technology including programming, databases, networking, and software engineering principles.
PSO 2	The Learner will be able to apply technical skills in software development, system analysis, and design using contemporary tools and technologies.
PSO 3	The Learner will able to have proficiency in identifying, formulating, and solving IT-related problems using appropriate techniques, algorithms, and methodologies.
PSO 4	The Learner will be able to have understanding of project management principles and methodologies relevant to IT projects, including planning, scheduling, and resource management
PSO 5	The Learner will be able to have effective communication skills, both oral and written, necessary for articulating technical concepts and collaborating in a team environment.

Semester – I		
NAMEOF THE COURSE	IMPERATIVE PROGRAMMING	
CLASS	FYBSc IT	
COURSE CODE	SBTTEC101	
NUMBER OF CREDITS	2	
NUMBER OF LECTURES PER WEEK	5	
TOTAL NUMBR OF LECTURES PER SEMESTER	75	
EVALUATION METHOD	INTERNAL ASSESSMENT	SEMESER END EXAMINATION
TOTAL MARKS	25	75
PASSING MARKS	10	30

COURSE OBJECTIVES:

CO 1.	The course aims to train the student to the basic concepts of the C-programming language.
CO 2.	It aims to train the students to understand the concept of conditional statement, loop, nested loop and break a large problem into smaller parts as a module or function.
CO 3.	It aims to train the students to understand the concept of string and be able to use anarray.to store multiple pieces of homogeneous data
CO 4.	It aims to train the students to understand the concept of pointer, and use a structure to storemultiple pieces of heterogeneous data.
CO 5.	This course involves a lab component which is designed to give the student hands-onexperience with the concepts.

COURSE LEARNING OUTCOMES:

CLO 1.	Read, understand and trace the execution of programs in C language.
CLO 2.	Draw flowchart and write the C code for a given algorithm.
CLO 3.	Implement the concept of control statements, loops, and functions to write a C program.
CLO 4.	Implement Programs with pointers and arrays, perform pointer arithmetic, and use the pre-processor.
CLO 5.	Implement Programs with structures and union

UNIT 1	**INTRODUCTION AND FUNDAMENTALS (15 LECTURES)**
1.1	**Introduction:** Types of Programming languages, History, features and application.Simple program logic, program development cycle, pseudocode statements andflowchart symbols, sentinel value to end a program, programming and userenvironments, evolution of programming models., desirable program characteristics.
1.2	**Fundamentals:** Structure of a program. Compilation and Execution of a Program, Character Set, identifiers and keywords, data types, constants, variables and arrays, declarations,expressions, statements, Variable definition, symbolic constants.
UNIT 2	**OPERATORS, EXPRESSIONS, DATA INPUT AND OUTPUT:** **(15 LECTURES)**
2.1	**Operators and Expressions:** Arithmetic operators, unary operators, relational and logical operators, assignment operators, assignment operators, the conditional operator, library functions.
2.2	**Data Input and output:** Single character input and output, entering input data, scanf function, printf function, gets and puts functions, interactive programming.
UNIT 3	**CONDITIONAL STATEMENTS, LOOPS AND FUNCTIONS:** **(15 LECTURES)**
3.1	**Conditional Statements and Loops:** Decision Making Within A Program, Conditions, Relational Operators, Logical Connectives, If Statement, If-Else Statement, Loops: While Loop, Do While, For Loop. Nested Loops, Infinite Loops, Switch Statement.
3.2	**Functions:** Overview, defining a function, accessing a function, passing arguments to a function, specifying argument data types, function prototypes, recursion, modular programming and functions, standard library of c functions, prototype of a function: foo1lal parameter list, return type, function call, block structure, passing arguments to a function: call by reference, call by value.

UNIT 4	**Program structure, Preprocessor, Arrays: (15 LECTURES)**
4.1	**Program structure:** Storage classes, automatic variables, external variables, static variables, multi file programs, more library functions,
4.2	**Preprocessor:** Features, #define and #include, Directives and Macros **Arrays:** Definition, processing, passing arrays to functions, multidimensional arrays, arrays and strings.
UNIT 5	**Pointers, Structures and Unions: (15 LECTURES)**
5.1	**Pointers:** Fundamentals, declarations, Pointers Address Operators, Pointer Type Declaration, Pointer Assignment, Pointer Initialization, Pointer Arithmetic, Functions and Pointers, Arrays And Pointers, Pointer Arrays, passing functions to other functions
5.2	**Structures and Unions:** Structure Variables, Initialization, Structure Assignment, Nested Structure, Structures and Functions, Structures and Arrays: Arrays of Structures, Structures Containing Arrays, Unions**,** Structures and pointers.

REFERENCES:

- Let us C, Yashavant Kanetkar, BPB Publications
- Programming in C, 3RD Edition, Ashok N. Kamthane & Amit Ashok Kamthane, Pearson India Education Services
- Head first C, David Griffiths & Dawn Griffiths, Shroff Publishers & Distributors

NAME OF THE COURSE	IMPERATIVE PROGRAMMING PRACTICAL	
CLASS	FYBSCIT	
COURSE CODE	SBTTECP101	
NUMBER OF CREDITS	2	
NUMBER OF LECTURES PER WEEK	3	
TOTAL NUMBER OF LECTURES PER SEMESTER	45	
EVALUATION METHOD	INTERNAL ASSESSMENT	SEMESTER END EXAMINATION
TOTAL MARKS	---	50
PASSING MARKS	---	20

List of Practical: (Can be done in any imperative language)	
1.	**Basic Programs:**
1.1.	Write a program to display the message HELLO WORLD.
1.2.	Write a program to declare some variables of type int, float and double. Assign some values to these variables and display these values.
1.3.	Write a program to find the addition, subtraction, multiplication and division of two numbers.
2.	**Programs on variables:**
2.1.	Write a program to swap two numbers without using third variable.
2.2.	Write a program to find the area of rectangle, square and circle.
2.3.	Write a program to find the volume of a cube, sphere, and cylinder.
3.	**Conditional statements and loops(basic)**
3.1.	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.
3.2.	Write a program to check whether the number is even or odd.
3.3.	Write a program to check whether the number is positive, negative or zero.
3.4.	Write a program to find the factorial of a number.

3.5.	Write a program to check whether the entered number is prime or not.
3.6.	Write a program to find the largest of three numbers.
4.	**Conditional statements and loops(advanced)**
4.1.	Write a program to find the sum of squares of digits of a number.
4.2.	Write a program to reverse the digits of an integer.
4.3.	Write a program to find the sum of numbers from 1 to 100.
4.4.	Write a program to print the Fibonacci series.
4.5.	Write a program to find the reverse of a number.
4.6.	Write a program to find whether a given number is palindrome or not.
4.7	Write a program that solve the quadratic equation $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$
4.8	Write a program to check whether the entered number is Armstrong or not.
4.9	Write a program to count the digit in a number
5.	**Programs on patterns:**
5.1.	Programs on different patterns.
6.	**Functions:**
6.1.	Programs on Functions.
7.	**Recursive functions**
7.1.	Write a program to find the factorial of a number using recursive function.
7.2.	Write a program to find the sum of natural number using recursive function.
8.	**Arrays**
8.1.	Write a program to find the largest value that is stored in the array.
8.2.	Write a program using pointers to compute the sum of all elements stored in an array.
8.3.	Write a program to arrange the ‘n’ numbers stored in the array in ascending and descending order.
8.4.	Write a program that performs addition and subtraction of matrices.
8.5.	Write a program that performs multiplication of matrices.
9.	**Pointers**

9.1	Write a program to demonstrate the use of pointers.
9.2.	Write a program to perform addition and subtraction of two pointer variables.
10.	**Structures and Unions**
10.1	Programs on structures.
10.2	Programs on unions.

Semester – I		
NAMEOF THE COURSE	DIGITAL ELECTRONICS	
CLASS	FYBSc IT	
COURSE CODE	SBTTEC102	
NUMBER OF CREDITS	2	
NUMBER OF LECTURES PER WEEK	5	
TOTAL NUMBR OF LECTURES PER SEMESTER	75	
EVALUATION METHOD	INTERNAL ASSESSMENT	SEMESER END EXAMINATION
TOTAL MARKS	25	75
PASSING MARKS	10	30

COURSE OBJECTIVES:

CO 1.	To acquire the basic knowledge of digital logic levels and application of knowledge to understand digital electronics circuits.
CO 2.	To introduce the basic concepts and laws involved in the Boolean algebra and logic families and digital circuits
CO 3.	To familiarize with the different number systems, logic gates, and combinational and sequential circuits utilized in the different digital circuits andsystems.
CO 4.	The course will help in design and analysis of the digital circuit and system
CO 5.	The course will help in design and analysis of counters and shift registers

COURSE LEARNING OUTCOMES:

CLO 1.	Gain knowledge between different types of number systems, and their conversions.
CLO 2.	Design various logic gates and simplify Boolean equations.
CLO 3.	To design and implement combinational logic & arithmetic circuits.

CLO 4.	Design various flip flops, conversion from one type of flip-flop to another CLO 5 Design different types of counters and shift registers.
CLO 5.	Design different types of counters and shift registers.

UNIT 1	NUMBER SYSTEM, BINARY ARITHMETIC (15 LECTURES)
1.1	**Number System:** Analog System, digital system, numbering system, binary number system, octal number system, hexadecimal number system, conversion from one number system to another, 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.
1.2	**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.
UNIT 2	**BOOLEAN ALGEBRA AND LOGIC GATES, MINTERM, MAXTERM AND KARNAUGH MAPS : (15 LECTURES)**
2.1	**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.
2.2	**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.
UNIT 3	**COMBINATIONAL LOGIC CIRCUITS, ARITHMETIC CIRCUITS: (15 LECTURES)**
3.1	**Combinational Logic Circuits:** Introduction, Multi-input, multi-output Combinational circuits, Code converters design and implementations
3.2	**Arithmetic Circuits:** Introduction, Adder, BCD Adder, Excess – 3 Adder, Binary Subtractors, BCD Subtractor, Multiplier, Comparator.
UNIT 4	**MULTIPLEXER, DEMULTIPLEXER, ALU,ENCODER AND DECODER SEQUENTIAL CIRCUITS: FLIP-FLOP (15 LECTURES))**
4.1	**Multiplexer, Demultiplexer, Alu, Encoder And Decoder:** Introduction, Multiplexer, Demultiplexer, Decoder, ALU, Encoders.
4.2	**Sequential Circuits: Flip-Flop:** Introduction, Terminologies used, S-R flip-flop, D flip-fop, J-K flip flop, Race-around condition, Master–slave JK flip-flop, T flip-flop, conversion from one type of flip-flop to another, Application of flip-flops
UNIT 5	**COUNTERS, SHIFT REGISTER: (15 LECTURES)**
5.1	**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.
5.2	**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.

REFERENCES:

- Digital Electronics and Logic Design, N. G. Palan ,Technova, 2nd revised edition
- Fundamentals of digital logic with Verilog Design, 2nd Edition, Brown, Stephen & Vranesic, Zvonko
- Digital circuits and design, 5th edition, 2018, Salivahanan S. & Arivazhagan S.

NAME OF THE COURSE	DIGITAL ELECTRONICS PRACTICAL	
CLASS	FYBSCIT	
COURSE CODE	SBTTECP102	
NUMBER OF CREDITS	2	
NUMBER OF LECTURES PER WEEK	3	
TOTAL NUMBER OF LECTURES PER SEMESTER	45	
EVALUATION METHOD TOTAL MARKS PASSING MARKS	INTERNAL ASSESSMENT --- ---	SEMESTER END EXAMINATION 50 20

List of Practical	
1.	**Study of Logic gates and their ICs and universal gates:**
1.1.	Study of AND, OR, NOT, XOR, XNOR, NAND and NOR gates
1.2.	IC 7400, 7402, 7404, 7408, 7432, 7486, 74266
1.3.	Implement AND, OR, NOT, XOR, XNOR using NAND gates.
1.4	Implement AND, OR, NOT, XOR, XNOR using NOR gates.
2.	**Implement the given Boolean expressions using minimum number of gates.**
2.1.	Verifying De Morgan's laws.
2.2.	Implement other given expressions using minimum number of gates.
2.3.	Implement other given expressions using minimum number of ICs.
3.	**Implement combinational circuits.**
3.1.	Design and implement combinational circuit based on the problem given and minimizing using K-maps.
4.	**Implement code converters.**

4.1.	Design and implement Binary – to – Gray code converter.
4.2.	Design and implement Gray – to – Binary code converter.
4.3.	Design and implement Binary – to – BCD code converter
4.4.	Design and implement Binary – to – XS-3 code converter
5.	**Implement Adder and Subtractor Arithmetic circuits.**
5.1.	Design and implement Half adder and Full adder.
5.2.	Design and implement BCD adder.
5.3.	Design and implement XS – 3 adder.
5.4.	Design and implement binary subtractor.
5.5.	Design and implement BCD subtractor.
5.6.	Design and implement XS – 3 subtractor.
6.	**Implement Arithmetic circuits.**
6.1	Design and implement a 2-bit by 2-bit multiplier.
6.2	Design and implement a 2-bit comparator.
7.	**Implement Encode and Decoder and Multiplexer and Demultiplexers.**
7.1	Design and implement 8:3 encoder.
7.2	Design and implement 3:8 decoder.
7.3	Design and implement 4:1 multiplexer. Study of IC 74153, 74157
7.4	Design and implement 1:4 demultiplexer. Study of IC 74139
7.5	Implement the given expression using IC 74151 8:1 multiplexer.
7.6	Implement the given expression using IC 74138 3:8 decoder.
8.	**Study of flip-flops and counters.**
8.1.	Study of IC 7473.
8.2.	Study of IC 7474.
8.3.	Study of IC 7476.
8.4.	Conversion of Flip-flops.
8.5.	Design of 3-bit synchronous counter using 7473 and required gates.
8.6	Design of 3-bit ripple counter using IC 7473.
9.	**Study of counter ICs and designing Mod-N counters.**
9.1	Study of IC 7490, 7492, 7493 and designing mod-n counters using these.

9.2	Designing mod-n counters using IC 7473 and 7400 (NAND gates)
10.	**Design of shift registers and shift register counters.**
10.1	Design serial – in serial – out, serial – in parallel – out, parallel – in serial – out, parallel – in parallel – out and bidirectional shift registers using IC 7474.
10.2	Study of ID 7495.
10.3	Implementation of digits using seven segment displays.

Semester – I		
NAMEOF THE COURSE	OPERATING SYSTEMS	
CLASS	FYBSc IT	
COURSE CODE	SBTTEC103	
NUMBER OF CREDITS	2	
NUMBER OF LECTURES PER WEEK	5	
TOTAL NUMBR OF LECTURES PER SEMESTER	75	
EVALUATION METHOD	INTERNAL ASSESSMENT	SEMESER END EXAMINATION
TOTAL MARKS	25	75
PASSING MARKS	10	30

COURSE OBJECTIVES:

CO 1.	To understand the services provided by and the design of an operating system
CO 2.	To make aware of different types of Operating System and their services.
CO 3.	To understand what a process is and learn different process scheduling algorithms and synchronization techniques to achieve better performance of a computer system.
CO 4.	To understand different approaches to memory management.
CO 5.	To understand the structure and organization of the file system.

COURSE LEARNING OUTCOMES:

CLO 1.	Explain the role of the operating system as a high-level interface to the hardware.
CLO 2.	Use OS as a resource manager that supports multiprogramming.
CLO 3.	Understands the different services provided by Operating System at different level.

CLO 4.	Understands the use of different process scheduling algorithm and synchronization techniques to avoid deadlock
CLO 5.	Understands the different services provided by Operating System at different level

UNIT 1	**INTRODUCTION, PROCESSES AND THREADS: (15 LECTURES)**
1.1	**Introduction:** What is an operating system? History of operating system, computer hardware, different operating systems, operating system concepts, system calls, operating system structure.
1.2	**Processes and Threads:** Processes, threads, inter process communication, scheduling, IPC problems.
UNIT 2	**MEMORY MANAGEMENT, FILE SYSTEMS :(15 LECTURES)**
2.1	**Memory Management:** No memory abstraction, memory abstraction: address spaces, virtual memory, page replacement algorithms, design issues for paging systems, implementation issues, segmentation.
2.2	**File Systems:** Files, directories, file system implementation, file-system management and optimization, MS-DOS file system, UNIX V7 file system, CD ROM file system.
UNIT 3	**INPUT-OUTPUT, DEADLOCKS :(15 LECTURES)**
3.1	**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,
3.2	**Deadlocks:** Resources, introduction to deadlocks, the ostrich algorithm, deadlock detection and recovery, deadlock avoidance, deadlock prevention, issues.
UNIT 4	**VIRTUALIZATION AND CLOUD, MULTIPLE PROCESSOR SYSTEMS: (15 LECTURES)**
4.1	**Virtualization and Cloud:**

	History, requirements for virtualization, type 1 and 2 hypervisors, techniques for efficient virtualization, hypervisor microkernels, memory virtualization, I/O virtualization, Virtual appliances, virtual machines on multicore CPUs, Clouds.
4.2	**Multiple Processor Systems** Multiprocessors, multicomputer, distributed systems.
UNIT 5	**CASE STUDY ON LINUX ANDANDROID, CASE STUDY ON WINDOWS: (15 LECTURES)**
5.1	**Case Study on LINUX and ANDROID:** History of Unix and Linux, Linux Overview, Processes in Linux, Memory management in Linux, I/O in Linux, Linux file system, security in Linux. Android
5.2	**Case Study on Windows:** History of windows through Windows 10, programming windows, system structure, processes and threads in windows, memory management, caching in windows, I/O in windows, Windows NT file system, Windows power management, Security in windows.

REFERENCES:

- Operating System Concepts – 8th Edition by Silberschatz, Abraham and others.
- Operating systems – 3rd Edition by Godbole, Kahate, Atul and Achyut S.
- Operating Systems: A concept based approach – 3rd Edition by Dhamdhare, Dhananjay M.

NAME OF THE COURSE	OPERATING SYSTEMS PRACTICAL	
CLASS	FYBSCIT	
COURSE CODE	SBTTECP103	
NUMBER OF CREDITS	2	
NUMBER OF LECTURES PER WEEK	3	
TOTAL NUMBER OF LECTURES PER SEMESTER	45	
EVALUATION METHOD TOTAL MARKS PASSING MARKS	INTERNAL ASSESSMENT --- ---	SEMESTER END EXAMINATION 50 20

List of Practical	
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:**
4.1	pwd, cd, absolute and relative paths, ls, mkdir, rmdir,
4.2	file, touch, rm, cp. mv, rename, head, tail, cat, tac, more, less, strings, chmod
5.	**Linux commands: Working with files:**
5.1.	ps, top, kill, pkill, bg, fg,
5.2.	grep, locate, find, locate.
5.3.	date, cal, uptime, w, whoami, finger, uname, man, df, du, free, whereis, which.
5.4.	Compression: tar, gzip.
6.	**Windows (DOS) Commands – 1**
6.1.	Date, time, prompt, md, cd, rd, path.
6.2.	Chkdsk, copy, xcopy, format, fidsk, cls, defrag, del, move.
7.	**Windows (DOS) Commands – 2**
7.1.	Diskcomp, diskcopy, diskpart, doskey, echo
7.2.	Edit, fc, find, rename, set, type, ver
8.	**Working with Windows Desktop and utilities**

8.1.	Notepad
8.2.	Wordpad
8.3.	Paint
8.4.	Taskbar
8.5.	Adjusting display resolution
8.6.	Using the browsers
8.7.	Configuring simple networking
8.8.	Creating users and shares
9.	**Working with Linux Desktop and utilities**
9.1.	The vi editor.
9.2.	Graphics
9.3.	Terminal
9.4.	Adjusting display resolution
9.5.	Using the browsers
9.6.	Configuring simple networking
9.7.	Creating users and shares
10.	**Installing utility software on Linux and Windows**

Semester – I		
NAMEOF THE COURSE	DISCRETE MATHEMATICS	
CLASS	FYBSc IT	
COURSE CODE	SBTTEC104	
NUMBER OF CREDITS	2	
NUMBER OF LECTURES PER WEEK	5	
TOTAL NUMBR OF LECTURES PER SEMESTER	75	
EVALUATION METHOD	INTERNAL ASSESSMENT	SEMESER END EXAMINATION
TOTAL MARKS	25	75
PASSING MARKS	10	30

COURSE OBJECTIVES:

CO 1.	The course objective is to provide students with an overview of discrete mathematics. To introduce the concepts of mathematical logic and set theory.
CO 2.	To introduce the concepts of logic, quantifiers, conditional propositions and Elementary Number Theory.
CO 3.	To learn various concepts like Mathematical Induction, sequences and recurrence relations.
CO 4.	To demonstrate Relations on Sets, Reflexivity, Symmetry, and Transitivity property. Understanding basic concepts and properties related to Graphs and Trees.
CO 5.	To acquire the basic knowledge of Probability Axioms, Conditional Probability, Multiplication rule and Independent Events.

COURSE LEARNING OUTCOMES:

CLO 1.	To perform the operations associated with sets, functions, and relations. Construct truth tables for expressions involving the logical connectives. Determine if a logical argument is valid or invalid.
CLO 2.	Construct correct direct and indirect proofs involving elementary number theory. Use a counterexample to show that a proposed statement involving concepts from elementary number theory is false. State and explain the Quotient Remainder Theorem (Division Algorithm).
CLO 3.	State the Principle of Mathematical Induction. Define and use the terms related to functions. Explain one-to-one, onto functions. Students will be able to solve problems based on each concept.
CLO 4.	State and explain binary relation, reflexive, symmetric, transitive, equivalence relations. Explain Properties of Graphs, Trees and use graph theory for solving problems.
CLO 5.	Apply principles of Probability, Permutations to solve various problems

UNIT 1	**INTRODUCTION, SET THEORY, THE LOGIC OF COMPOUND STATEMENTS: (15 LECTURES)**
1.1	**Introduction**: Variables, The Language of Sets, The Language of Relations and Function
1.2	**Set Theory:** Definitions and the Element Method of Proof, Properties of Sets, Disproofs, Algebraic Proofs, Boolean Algebras, Russell's Paradox and the Halting Problem.
1.3	**The Logic of Compound Statements**: Logical Form and Logical Equivalence, Conditional Statements, Valid and Invalid Arguments
UNIT 2	**QUANTIFIED STATEMENTS, ELEMENTARY NUMBER THEORY AND METHODS OF PROOF :(15 LECTURES)**
2.1	**Quantified Statements**: Predicates and Quantified Statements, Statements with Multiple Quantifiers, Arguments with Quantified Statements.

2.2	**Elementary Number Theory and Methods of Proof:** Introduction to Direct Proofs, Rational Numbers, Divisibility, Division into Cases and the Quotient-Remainder Theorem, Applications in algorithms.
UNIT 3	**SEQUENCES, MATHEMATICAL INDUCTION,AND RECURSION, FUNCTIONS: (15 LECTURES)**
3.1	**Sequences, Mathematical Induction, and Recursion**: Sequences, Mathematical Induction, Strong Mathematical Induction and the WellOrdering Principle for the Integers, Correctness of algorithms, defining sequences recursively, solving recurrence relations by iteration, Second order linear homogenous recurrence relations with constant coefficients. General recursive definitions and structural induction.
3.2	Functions: Functions Defined on General Sets, One-to-One and Onto, Inverse Functions, Composition of Functions, Cardinality with Applications to Computability
UNIT 4	**RELATIONS, GRAPHS AND TREES :(15 LECTURES)**
4.1	**Relations**: Relations on Sets, Reflexivity, Symmetry, and Transitivity, Equivalence Relations.
4.2	**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.
UNIT 5	**COUNTING AND PROBABILITY : (15 LECTURES)**
5.1	**Counting and Probability:** Introduction, Possibility Trees and the Multiplication Rule, Possibility Trees and the Multiplication Rule,
5.2	Probability Axioms and Expected Value, Conditional Probability, Bayes' Formula, and Independent Events.

REFERENCES:

- Discrete Mathematics with Applications Sussana S. Epp Cengage Learning 4th2010
- Elements of discrete mathematics: A computer oriented approach. 4th ed. Liu, C.L. & Mohapatra D.P.
- Discrete Mathematics for computer scientists and mathematicians 2nd ed.Mott, Joe L

NAME OF THE COURSE	DISCRETE MATHEMATICS PRACTICAL	
CLASS	FYBSCIT	
COURSE CODE	SBTTECP104	
NUMBER OF CREDITS	2	
NUMBER OF LECTURES PER WEEK	3	
TOTAL NUMBER OF LECTURES PER SEMESTER	45	
EVALUATION METHOD	INTERNAL ASSESSMENT	SEMESTER END EXAMINATION
TOTAL MARKS	---	50
PASSING MARKS	---	20

List of Practical: Write the programs for the following using SCILAB	
1.	**Set Theory**
1.1	Inclusion Exclusion principle.
1.2	Power Sets
1.3	Mathematical Induction
2.	**Functions and Algorithms**
2.1.	Recursively defined functions
2.2.	Cardinality
2.3.	Polynomial evaluation
2.4.	Greatest Common Divisor
3.	**Counting**
3.1.	Sum rule principle
3.2.	Product rule principle
3.3.	Factorial
3.4.	Binomial coefficients
3.5.	Permutations
3.6.	Permutations with repetitions
3.7.	Combinations
3.8.	Combinations with repetitions

3.9	Ordered partitions
3.10	Unordered partitions
4.	**Probability Theory**
4.1.	Sample space and events
4.2.	Finite probability spaces
4.3.	Equiprobable spaces
4.4.	Addition Principle
4.5.	Conditional Probability
4.6.	Multiplication theorem for conditional probability
4.7.	Independent events
4.8.	Repeated trials with two outcomes
5.	**Graph Theory**
5.1	Paths and connectivity
5.2	Minimum spanning tree
5.3	Isomorphism
6.	**Directed Graphs**
6.1	Adjacency matrix
6.2	Path matrix
7.	**Properties of integers**
7.1.	Division algorithm
7.2.	Primes
7.3.	Euclidean algorithm
7.4.	Fundamental theorem of arithmetic
7.5.	Congruence relation
7.6.	Linear congruence equation
8.	**Algebraic Systems**
8.1.	Properties of operations
8.2.	Roots of polynomials
9.	**Boolean Algebra**
9.1.	Basic definitions in Boolean Algebra

9.2.	Boolean algebra as lattices
10.	**Recurrence relations**
10.1.	Linear homogeneous recurrence relations with constant coefficients
10.2.	Solving linear homogeneous recurrence relations with constant coefficients
10.3.	Solving general homogeneous linear recurrence relations

Semester – I		
NAMEOF THE COURSE	COMMUNICATION SKILLS	
CLASS	FYBSc IT	
COURSE CODE	SBTTEC105	
NUMBER OF CREDITS	2	
NUMBER OF LECTURES PER WEEK	5	
TOTAL NUMBR OF LECTURES PER SEMESTER	75	
EVALUATION METHOD	INTERNAL ASSESSMENT	SEMESER END EXAMINATION
TOTAL MARKS	25	75
PASSING MARKS	10	30

COURSE OBJECTIVES:

CO 1.	The course is intended to emphasize the essential aspects of effective written and verbal communication necessary for professional success. Familiarize students with the basics of Technology-enabled Business Communication.
CO 2.	The course is designed to empower students to carry out day to day communication at the work place. To impart adequate understanding of various types of communication to facilitate efficient interpersonal communication. To impart the correct practices and strategies of Effective Business writing
CO 3.	This course is designed to develop the skills of the students in preparing for job search and negotiating their use in GDs and interviews. Helps students in Communicating across Functional Areas.
CO 4.	Understanding Ethics in Business Communication and Business Communication Aids. CO 5. The course is designed to develop competence in communication skills related to production & presentation of messages in multiple formats & understand the importance of body language.

COURSE LEARNING OUTCOMES:

CLO 1.	The students should be able to: Apply Verbal and Non-Verbal Communication Techniques in the Professional Environment. Students will be able to deliver messages that incorporate the appropriate use of organizing content, language, vocabulary, kinesics, eye contact, appearance, visual aids, and time constraints.
CLO 2.	Demonstrate ability to interpret texts and observe the rules of good writing. Prepare and present effective presentations aided by ICT tools. The student will be able to write an impressive resume and face the interview confidently. CLO 3. The students will be able to apply good Oral Communication Skills for Business purpose like in meetings, conferences, GDs etc.
CLO 3.	The students will be able to apply good Oral Communication Skills for Business purpose like in meetings, conferences, GDs etc.
CLO 4.	The Student will be able to conduct themselves using proper business ethics and will be able to use various Business Communication Aids
CLO 5.	The student will be able to write impressive official correspondence and also learn to make and give effective presentations.

UNIT 1	**THE SEVEN CS OF EFFECTIVE COMMUNICATION UNDERSTANDING BUSINESS COMMUNICATION: (15 LECTURES)**
1.1	**The Seven Cs of Effective Communication:** Completeness, Conciseness, Consideration, Concreteness, Clarity, Courtesy, Correctness
1.2	**Understanding Business Communication:** Nature and Scope of Communication, Non-verbal Communication, Cross- cultural communication, Technology-enabled Business Communication
UNIT 2	**WRITING BUSINESS MESSAGES AND DOCUMENTS, DEVELOPING ORAL COMMUNICATION SKILLS FOR BUSINESS: (15 LECTURES)**
2.1	**Writing Business Messages and Documents:**

	Business writing, Business Correspondence, Instructions Business Reports and Proposals, Career building and Resume writing.
2.2	**Developing Oral Communication Skills for Business:** Effective Listening, Business Presentations and Public Speaking, Conversations, Interviews
UNIT 3	**DEVELOPING ORAL COMMUNICATIONSKILLS FOR BUSINESS UNDERSTANDING SPECIFIC COMMUNICATION NEEDS :(15 LECTURES)**
3.1	**Developing Oral Communication Skills for Business:** Meetings and Conferences, Group Discussions and Team Presentations, Team Briefing,
3.2	**Understanding Specific Communication Needs:** Communication across Functional Areas
UNIT 4	**UNDERSTANDING SPECIFIC COMMUNICATION NEEDS : (15 LECTURES)**
4.1	**Understanding Specific Communication Needs:** Corporate Communication, Persuasive Strategies in Business Communication, Ethics in Business Communication, Business Communication Aids
UNIT 5	**PRESENTATION PROCESS :(15 LECTURES)**
5.1	**Presentation Process:** 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

.

REFERENCES:

- Business Communication Meenakshi Raman and Prakash Singh Oxford University Press 2nd ed.
- Basic Business Communication: Making connections in a digital world. 11th ed. Lesikar Raymond V

- Professional Communication Koneru, Aruna
- Business correspondence and report writing : A poractical approach to business & technical communication. 4th ed.

NAME OF THE COURSE	COMMUNICATION SKILLS PRACTICAL	
CLASS	FYBSCIT	
COURSE CODE	SBTTECP105	
NUMBER OF CREDITS	2	
NUMBER OF LECTURES PER WEEK	3	
TOTAL NUMBER OF LECTURES PER SEMESTER	45	
EVALUATION METHOD	INTERNAL ASSESSMENT	SEMESTER END EXAMINATION
TOTAL MARKS	---	50
PASSING MARKS	---	20

List of Practical Questions:	
1.	Communication Origami, Guessing Game, Guessing the emotion
2.	Body Language, Follow All Instructions, Effective Feedback Skills
3.	The Name Game, Square Talk (Effective Communication), Room 101 (Influential and persuasive skills)
4.	Back to Back Communication, Paper Shapes (Importance of two-way communication), Memory Test(Presentation Skills)
5.	Exercises on Communication Principles
6.	Exercises on communication icebreakers
7.	Communication exercises
	For the following practicals, Microsoft Office, Open Office, Libre Office or any other software suite can be used.
8.	Use of word processing tools for communication
9.	Use of spreadsheet tools for communication
10.	Use of presentation tools for communication

SEMESTER II

Semester – II		
NAMEOF THE COURSE	OBJECT ORIENTED PROGRAMMING	
CLASS	FYBSc IT	
COURSE CODE	SBTTEC201	
NUMBER OF CREDITS	2	
NUMBER OF LECTURES PER WEEK	5	
TOTAL NUMBR OF LECTURES PER SEMESTER	75	
EVALUATION METHOD	INTERNAL ASSESSMENT	SEMESER END EXAMINATION
TOTAL MARKS	25	75
PASSING MARKS	10	30

COURSE OBJECTIVES:

CO 1.	The objective of the course is to teach the basic concepts and techniques which form the object-oriented programming paradigm.
CO 2.	To learn the concept of class and object using C++ and develop classes for simple applications.
CO 3.	To learn the concept of Constructors and destructors in C++. program.
CO 4.	To learn the concept of function overloading, operator overloading, virtual functions and polymorphism.
CO 5.	Classify inheritance with the understanding of early and late binding, usage of exception handling.

COURSE LEARNING OUTCOMES:

CLO 1.	Creating simple programs using classes and objects in C++.
CLO 2.	Implement programs using constructors, destructors and operator overloading.

CLO 3.	Apply fundamental algorithmic problems including polymorphism and virtual function.
CLO 4.	Implement Object Oriented Programs using the concept of inheritance and exceptional handling.
CLO 5.	Implement Object Oriented Programs using templates and file handling concepts.

UNIT 1	**OBJECT ORIENTED METHODOLOGY, PRINCIPLES OF OOPS : (15 LECTURES)**
1.1	**Object Oriented Methodology**: Introduction, Advantages and Disadvantages of Procedure Oriented Languages, what is Object Oriented? What is Object Oriented Development? Object Oriented Themes, Benefits and Application of OOPS.
1.2	**Principles of OOPS:** OOPS Paradigm, Basic Concepts of OOPS: Objects, Classes, Data Abstraction and Data Encapsulation, Inheritance, Polymorphism, Dynamic Binding, Message Passing
UNIT 2	**CLASSES AND OBJECTS, CONSTRUCTORS AND DESTRUCTORS: (15 LECTURES)**
2.1	**Classes and Objects:** Simple classes (Class specification, class members accessing), Defining member functions, passing object as an argument, Returning object from functions, friend classes, Pointer to object, Array of pointer to object.
2.2	**Constructors and Destructors:** Introduction, Default Constructor,Parameterized Constructor and examples, Destructors
UNIT 3	**POLYMORPHISM, VIRTUAL FUNCTIONS : (15 LECTURES)**
3.1	**Polymorphism:** Concept of function overloading, overloaded operators, overloading unary and binary operators, overloading comparison operator, overloading arithmetic assignment operator, Data Conversion between objects and basic types
3.2	**Virtual Functions:** Introduction and need, Pure Virtual Functions, Static Functions, this Pointer, abstract classes, virtual destructors.

UNIT 4	**PROGRAM DEVELOPMENT USING INHERITANCE, EXCEPTION HANDLING :(15 LECTURES)**
4.1	**Program development using Inheritance:** Introduction, understanding inheritance, Advantages provided by inheritance, choosing the access specifier, Derived class declaration, derived class constructors, class hierarchies, multiple inheritance, multilevel inheritance, containership, hybrid inheritance.
4.2	**Exception Handling:** Introduction, Exception Handling Mechanism, Concept of throw & catch with example
UNIT 5	**TEMPLATES, WORKING WITH FILES: (15 LECTURES)**
5.1	**Templates:** Introduction, Function Template and examples, Class Template and examples
5.2	**Working with Files:** Introduction, File Operations, Various File Modes, File Pointer and their Manipulation

REFERENCES:

- Object-oriented programming with C++., Balaguruswamy, E., Tata McGraw-Hill 1995
- Mastering C++., Venugopal, K.R. & Buyya, Rajkumar, McGraw Hill Education (India) 2013
- Let us C++. 2nd ed., Kanetkar, Yashavant, BPB Publications, 2010
- C++: The complete reference. 4th ed.Schildt, Herbert, Tata McGraw-Hill Publishing Co 2003

NAME OF THE COURSE	OBJECT ORIENTED PROGRAMMING PRACTICAL	
CLASS	FYBSCIT	
COURSE CODE	SBTTECP201	
NUMBER OF CREDITS	2	
NUMBER OF LECTURES PER WEEK	3	
TOTAL NUMBER OF LECTURES PER SEMESTER	45	
EVALUATION METHOD TOTAL MARKS	INTERNAL ASSESSMENT ---	SEMESTER END EXAMINATION 50
PASSING MARKS	---	20

List of Practical: To be implemented using object oriented language	
1.	**Classes and methods**
1.1	Design an employee class for reading and displaying the employee information, the getInfo() and displayInfo() methods will be used repectively. Where getInfo() will beprivate method
1.2	Design the class student containing getData() and displayData() as two of its methodswhich will be used for reading and displaying the student information respectively. Where getData() will be private method.
1.3	Design the class Demo which will contain the following methods: readNo(), factorial() for calculating the factorial of a number, reverseNo() will reverse the givennumber, isPalindrome() will check the given number is palindrome, isArmstrong() which will calculate the given number is armStrong or not.Where readNo() will be private method.
1.4	Write a program to demonstrate function definition outside class and accessing class members in function definition.
2.	**Using friend functions.**
2.1	Write a friend function for adding the two complex numbers, using a single class

2.2	Write a friend function for adding the two different distances and display its sum, using two classes.
2.3	Write a friend function for adding the two matrix from two different classes and display its sum.
3.	**Constructors and method overloading.**
3.1	Design a class Complex for adding the two complex numbers and also show the use of constructor.
3.2	Design a class Geometry containing the methods area() and volume() and also overload the area() function .
3.3	Design a class StaticDemo to show the implementation of static variable and static function.
4.	**Operator Overloading**
4.1	Overload the operator unary(-) for demonstrating operator overloading.
4.2	Overload the operator + for adding the timings of two clocks, And also pass objects as an argument.
4.3	Overload the + for concatenating the two strings. For e.g “Py” + “thon” = Python
5.	**Inheritance**
5.1	Design a class for single level inheritance using public and private type derivation.
5.2	Design a class for multiple inheritance.
5.3	Implement the hierarchical inheritance.
6.	**Virtual functions and abstract classes**
6.1	Implement the concept of method overriding.
6.2	Show the use of virtual function
6.3	Show the implementation of abstract class.
7.	**String handling**
7.1	String operations for string length , string concatenation
7.2	String operations for string reverse, string comparison,
7.3	Console formatting functions.

8.	**Exception handling**
8.1	Show the implementation of exception handling
8.2	Show the implementation for exception handling for strings
8.3	Show the implementation of exception handling for using the pointers.
9.	**File handling**
9.1	Design a class FileDemo open a file in read mode and display the total number of words and lines in the file.
9.2	Design a class to handle multiple files and file operations
9.3	Design a editor for appending and editing the files
10.	**Templates**
10.1	Show the implementation for the following
10.2	Show the implementation of template class library for swap function.
10.3	Design the template class library for sorting ascending to descending and viceversa

Semester – II		
NAMEOF THE COURSE	MICROPROCESSOR ARCHITECUTURE	
CLASS	FYBSc IT	
COURSE CODE	SBTTEC202	
NUMBER OF CREDITS	2	
NUMBER OF LECTURES PER WEEK	5	
TOTAL NUMBR OF LECTURES PER SEMESTER	75	
EVALUATION METHOD	INTERNAL ASSESSMENT	SEMESER END EXAMINATION
TOTAL MARKS	25	75
PASSING MARKS	10	30

COURSE OBJECTIVES:

CO 1.	Students will be able to differentiate between Microprocessor, Microcontroller and Microcomputers and will also learn 8085 architectures.
CO 2.	Students will be able to write Assembly Language Programs and will learn about stacks, subroutines and interrupts.
CO 3.	Students will be able to distinguish between Counters and Timers and will study about stacks and subroutines.
CO 4.	Students will be able to do Code Conversion, BCD Arithmetic, and 16-Bit Data Operations and will learn about Programming Tools and Interrupts.
CO 5.	Students will be able to gain knowledge about Pentium and Pentium Pro microprocessors

COURSE LEARNING OUTCOMES:

CLO 1.	Compare Microprocessor and Microcontroller. Explain 8085 architecture.
CLO 2.	Writing Assembly Language Programs.
CLO 3.	Distinguish between Counters and Timers.
CLO 4.	State and explain the code conversions and various data operations.
CLO 5.	Explain Pentium and Pentium Pro microprocessors

UNIT 1	**MICROPROCESSOR, MICROCOMPUTERS, ANDASSEMBLY LANGUAGE, MICROPROCESSOR ARCHITECTURE AND MICROCOMPUTER SYSTEM 8085 MICROPROCESSOR ARCHITECTURE AND MEMORY INTERFACE :(15 LECTURES)**
1.1	**Microprocessor, microcomputers, and Assembly Language:** Microprocessor, Microprocessor Instruction Set and Computer Languages, From Large Computers to Single-Chip Microcontrollers, Applications.
1.2	**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.
1.3	**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.
UNIT 2	**INTERFACING OF I/O DEVICES INTRODUCTION TO 8085 ASSEMBLY LANGUAGE PROGRAMMING INTRODUCTION TO 8085 INSTRUCTIONS: (15 LECTURES)**
2.1	**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.
2.2	**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.
2.3	**Introduction to 8085 Instructions:**

	Data Transfer Operations, Arithmetic Operations, Logic Operation, Branch Operation, Writing Assembly Languages Programs, Debugging a Program.
UNIT 3	**PROGRAMMING TECHNIQUES WITHADDITIONAL INSTRUCTIONS COUNTERS AND TIME DELAYS STACKS AND SUB-ROUTINES :(15 LECTURES)**
3.1	**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.
3.2	**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.
3.3	**Stacks and Sub-Routines:** Stack, Subroutine, Restart, Conditional Call, Return Instructions, Advanced Subroutine concepts.
UNIT 4	**CODE CONVERSION, BCD ARITHMETIC,AND 16-BIT DATA OPERATIONS SOFTWARE DEVELOPMENT SYSTEM AND ASSEMBLERS INTERRUPTS (15 LECTURES)**
4.1	**Code Conversion, BCD Arithmetic, and 16-Bit Data Operations:**
	BCD-to-Binary Conversion, Binary-to-BCD Conversion, BCD-toSeven-Segment- LED Code Conversion, Binary-to-ASCII and ASCIIto-Binary Code Conversion, BCD Addition, BCD Subtraction, Introduction To Advanced Instructions and Applications, Multiplication, Subtraction With Carry.
4.2	**Software Development System and Assemblers:** Microprocessors-Based Software Development system, Operating System and Programming Tools, Assemblers and Cross-Assemblers, Writing Program Using Cross Assemblers.
4.3	**Interrupts:**

	The 8085 Interrupt, 8085 Vectored Interrupts, Restart as S/W Instructions, Additional I/O Concepts and processes
UNIT 5	**THE PENTIUM AND PENTIUM PROMICROPROCESSORS CORE 2 AND LATER MICROPROCESSORS SUN SPARC MICROPROCESSOR: (15 LECTURES)**
5.1	**The Pentium and Pentium Pro microprocessors:** Introduction, Special Pentium registers, Memory management, Pentium instructions, Pentium Pro microprocessor, Special Pentium Pro features.
5.2	**Core 2 and later Microprocessors:** Introduction, Pentium II software changes,Pentium IV and Core 2, i3, i5 and i7.
5.3	**SUN SPARC Microprocessor:** Architecture, Register file, data types and instruction format

REFERENCES:

- Microprocessors Architecture, Programming and Applications with the 8085 Ramesh Gaonkar PENRAM Fifth 2012
- The 8051 Microcontroller and Embedded Systems Muhammad Ali Mazidi Pearson Second 2011

NAME OF THE COURSE	MICROPROCESSOR ARCHITECUTURE PRACTICAL	
CLASS	FYBSCIT	
COURSE CODE	SBTTECP202	
NUMBER OF CREDITS	2	
NUMBER OF LECTURES PER WEEK	3	
TOTAL NUMBER OF LECTURES PER SEMESTER	45	
EVALUATION METHOD	INTERNAL ASSESSMENT	SEMESTER END EXAMINATION
TOTAL MARKS	---	50
PASSING MARKS	---	20

List of Practical	
1.	**Perform the following Operations related to memory locations.**
1.1	Store the data byte 32H into memory location 4000H.
1.2	Exchange the contents of memory locations 2000H and 4000H
2.	**Simple assembly language programs.**
2.1	Subtract the contents of memory location 4001H from the memory location 2000H and place the result in memory location 4002H.
2.2	Subtract two 8-bit numbers.
2.3	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.
2.4	Add the contents of memory locations 40001H and 4001H and place the result in the memory locations 4002Hand 4003H.
2.5	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 thetwo 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.
2.6	Find the l's complement of the number stored at memory location 4400H and store the complemented number at memory location 4300H.
2.7	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.**
3.1	Pack the two unpacked BCD numbers stored in memory locations 4200H and 4201Hand store result in memory location 4300H. Assume the least significant digit is stored at 4200H.
3.2	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.**
4.1	Write a program to shift an eight bit data four bits right. Assume that data is in register C.
4.2	Program to shift a 16-bit data 1 bit left. Assume data is in the HL register pair
4.3	Write a set of instructions to alter the contents of flag register in 8085.
4.4	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.**
5.1	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. b. Consider the sum to be 16 bit number. Store the sum at memory locations 4300H and 4301H
5.2	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.
5.3	Divide 16 bit number stored in memory locations 2200H and 2201H by the 8 bit number stored at memory location 2202H. Store the quotient in memory locations 2300H and 2301H and remainder in memory locations 2302H and 2303H.
5.4	Find the number of negative elements (most significant bit 1) in a block of data. The length of the block is in memory location 2200H and the block itself begins in memory location 2201H. Store the number of negative elements in memory location 2300H
5.5	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.**

6.1	Write a program to sort given 10 numbers from memory location 2200H in the ascending order.
6.2	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:
6.3	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.
6.4	Find the square of the given numbers from memory location 6100H and store the result from memory location 7000H
6.5	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
6.6	Two decimal numbers six digits each, are stored in BCD package form. Each numberoccupies a sequence of byte in the memory. The starting address of first number is 6000H Write an assembly language program that adds these two numbers and stores the sum in the same format starting from memory location 6200H
6.7	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.**
7.1	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 startingaddress of 50 number list is 2200H
7.2	Write assembly language program with proper comments for the following: A block of data consisting of 256 bytes is stored in memory starting at 3000H.

	This block is to be shifted (relocated) in memory from 3050H onwards. Do not shift the block or part of the block anywhere else in the memory.
7.3	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.
7.4	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
7.5	Write an assembly language program to generate fibonacci number.
7.6	Program to calculate the factorial of a number between 0 to 8.
8.	**String operations in assembly programs.**
8.1	Write an 8085 assembly language program to insert a string of four characters from the tenth location in the given array of 50 characters
8.2	Write an 8085 assembly language program to delete a string of 4 characters from the tenth location in the given array of 50 characters.
8.3	Multiply the 8-bit unsigned number in memory location 2200H by the 8-bit unsignednumber 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.
8.4	Divide the 16-bit unsigned number in memory locations 2200H and 2201H (most significant bits in 2201H) by the B-bit unsigned number in memory location 2300H store the quotient in memory location 2400H and remainder in 2401H
8.5	DAA instruction is not present. Write a sub routine which will perform the same task as DAA.
9.	**Calculations on memory locations.**
9.1	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 isindicated by writing 01H at port 10
9.2	Arrange an array of 8 bit unsigned no in descending order
9.3	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
9.4	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.
9.5	Write a simple program to Split a HEX data into two nibbles and store it in memory
10.	**Operations on BCD numbers.**
10.1	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.
10.2	Subtract the BCD number stored in E register from the number stored in the D register
10.3	Write an assembly language program to multiply 2 BCD numbers

Semester – II		
NAMEOF THE COURSE	WEB PROGRAMMING	
CLASS	FYBSc IT	
COURSE CODE	SBTTEC203	
NUMBER OF CREDITS	2	
NUMBER OF LECTURES PER WEEK	5	
TOTAL NUMBR OF LECTURES PER SEMESTER	75	
EVALUATION METHOD	INTERNAL ASSESSMENT	SEMESER END EXAMINATION
TOTAL MARKS	25	75
PASSING MARKS	10	30

COURSE OBJECTIVES:

CO 1.	Develop the ability to logically plan and develop web pages
CO 2.	Learn to write, test, and debug web pages using HTML and JavaScript
CO 3.	Understand the principles of creating an effective web page, including an in-depth consideration of information architecture.
CO 4.	Develop basic programming skills using JavaScript and Angular JS.
CO 5.	Become familiar with PHP & MySQL principles that relate to web design and learn how to implement theories into practice.

COURSE LEARNING OUTCOMES:

CLO 1.	Describe the concepts of World Wide Web, and the requirements of effective web design.
CLO 2.	Develop web pages using the HTML and CSS features with different layouts as per need of applications.
CLO 3.	To construct basic websites using HTML and Cascading Style Sheets.
CLO 4.	Use the JavaScript to develop the dynamic web pages.
CLO 5.	Construct simple web pages in PHP and MySQL.

UNIT 1	INTERNET AND THE WORLD WIDE WEB HTML5 :(15 LECTURES)
1.1	**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
1.2	**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.
UNIT 2	**HTML5 PAGE LAYOUT AND NAVIGATION, HTML5 TABLES, FORMS AND MEDIA :(15 LECTURES)**
2.1	**HTML5 Page layout and navigation:** Creating navigational aids: planning site organization, creating text based navigationbar, 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.
2.2	**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.

UNIT 3	**JAVA SCRIPT, CORE JAVASCRIPT PROPERTIES AND METHODS, EVENTS AND EVENT HANDLERS :(15 LECTURES)**
3.1	**Java Script:** Introduction, Client-Side JavaScript, Server-Side JavaScript, JavaScript Objects, JavaScript Security,
3.2	**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
3.3	**Statements:** Break, comment, continue, delete, do...while, export, for, for...in, function, if...else, import, labelled, return, switch, var, while, with
3.4	**Core JavaScript (Properties and Methods of Each) :** Array, Boolean, Date, Function, Math, Number, Object, String, regExp
3.5	**Document and its associated objects:** document, Link, Area, Anchor, Image, Applet, Layer
3.6	**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
UNIT 4	**PHP :(15 LECTURES)**
4	**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
UNIT 5	**ADVANCED PHP AND MYSQL :(15 LECTURES)**
5	**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

REFERENCES:

- Web Design The CompleteReference Thomas Powell Tata McGraw Hill -
- HTML5 Step by Step Faithe Wempen Microsoft Press 2011
- PHP 6 and MySQL Bible Steve Suehring,Tim Converse, Joyce Park Wiley 2009
- Head First HTML 5 programming Eric Freeman O'Reilly 2013
- JavaScript 2.0: The CompleteReference Thomas

NAME OF THE COURSE	WEB PROGRAMMING PRACTICAL	
CLASS	FYBSCIT	
COURSE CODE	SBTTECP203	
NUMBER OF CREDITS	2	
NUMBER OF LECTURES PER WEEK	3	
TOTAL NUMBER OF LECTURES PER SEMESTER	45	
EVALUATION METHOD	INTERNAL ASSESSMENT	SEMESTER END EXAMINATION
TOTAL MARKS	---	50
PASSING MARKS	---	20

List of Practical	
1.	**Use of Basic Tags**
1.1	Design a web page using different text formatting tags.
1.2	Design a web page with links to different pages and allow navigation between web pages.
1.3	Design a web page demonstrating all Style sheet types
2.	**Image maps, Tables, Forms and Media**
2.1	Design a web page with Imagemaps.
2.2	Design a web page demonstrating different semantics

2.3	Design a web page with different tables. Design a webpages using table so that the content appears well placed.
2.4	Design a web page with a form that uses all types of controls.
2.5	Design a web page embedding with multimedia features.
3.	**Java Script**
3.1	Using JavaScript design, a web page that prints factorial/Fibonacci series/any given series.
3.2	Design a form and validate all the controls placed on the form using Java Script.
3.3	Write a JavaScript program to display all the prime numbers between 1 and 100.
3.4	Write a JavaScript program to accept a number from the user and display the sum of its digits.
3.5	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).
3.6	Write a java script program to design simple calculator.
4.	**Control and looping statements and Java Script references**
4.1	Design a web page demonstrating different conditional statements.
4.2	Design a web page demonstrating different looping statements.
4.3	Design a web page demonstrating different Core JavaScript references (Array, Boolean, Date, Function, Math, Number, Object, String, regExp).
5.	**Basic PHP I**
5.1	Write a PHP Program to accept a number from the user and print it factorial.
5.2	Write a PHP program to accept a number from the user and print whether it is prime or not.
6.	**Basic PHP II**
6.1	Write a PHP code to find the greater of 2 numbers. Accept the no. from the user.
6.2	PHP program to display the following Binary Pyramid:1 0 1 1 0 1 0 1 0 1

7.	**String Functions and arrays**
7.1	Write a PHP program to demonstrate different string functions.
7.2	Write a PHP program to create one dimensional array.
8.	**PHP and Database**
8.1	Write a PHP code to create: Create a database College Create a table Department (Dname, Dno, Number_Of_faculty)
8.2	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.
8.3	Design a PHP page for authenticating a user.
9.	**Email**
9.1.	Write a program to send email with attachment.
10.	**Sessions and Cookies**
10.1	Write a program to demonstrate use of sessions and cookies.

Semester – II		
NAMEOF THE COURSE	NUMERICAL STATISTICAL METHODS	
CLASS	FYBSc IT	
COURSE CODE	SBTTEC204	
NUMBER OF CREDITS	2	
NUMBER OF LECTURES PER WEEK	5	
TOTAL NUMBR OF LECTURES PER SEMESTER	75	
EVALUATION METHOD	INTERNAL ASSESSMENT	SEMESER END EXAMINATION
TOTAL MARKS	25	75
PASSING MARKS	10	30

COURSE OBJECTIVES:

CO 1.	To develop the student's ability to deal with numerical and quantitative issues in business
CO 2.	To enable the use of statistical, graphical and algebraic techniques wherever relevant.
CO 3.	To have a proper understanding of Statistical applications in IT and Research industry.
CO 4.	Recover deleted files, hidden files, and temporary information that would be used as proof.
CO 5.	To understand the usage of correct tools for forensic investigations.

COURSE LEARNING OUTCOMES:

CLO 1.	Understand the various approaches dealing the data using theory of probability.
CLO 2.	Develop a framework for estimating and predicting the different sample of data for handling the uncertainties.
CLO 3.	Understand error, source of error and its effect on any numerical computation and also analyzing the efficiency of any numerical algorithm.

CLO 4.	Learn how to obtain numerical solution of nonlinear equations using Bisection, Newton – Raphson and Regula – Falsi method iteration methods.
CLO 5.	Solve system of linear equations numerically using direct and iterative methods

UNIT 1	**MATHEMATICAL MODELING AND ENGINEERING PROBLEM SOLVING APPROXIMATIONS AND ROUND-OFF ERRORS TRUNCATION ERRORS AND THE TAYLOR SERIES : (15 LECTURES)**
1.1	**Mathematical Modeling and Engineering Problem Solving:** A
	Simple Mathematical Model, Conservation Laws and Engineering Problems
1.2	**Approximations and Round-Off Errors:** Significant Figures, Accuracy andPrecision, Error Definitions, Round-Off Errors
1.3	**Truncation Errors and the Taylor Series:** The Taylor Series, Error Propagation, Total Numerical Errors, Formulation Errors and Data Uncertainty
UNIT 2	**SOLUTIONS OF ALGEBRAIC AND TRANSCENDENTAL EQUATIONS INTERPOLATION :(15 LECTURES)**
2.1	**Solutions of Algebraic and Transcendental Equations:** The Bisection Method, The Newton-Raphson Method, The Regula-falsi method, The Secant Method.
2.2	**Interpolation:** Forward Difference, Backward Difference, Newton's Forward Difference Interpolation, Newton's Backward Difference Interpolation, Lagrange'sInterpolation.
UNIT 3	**SOLUTION OF SIMULTANEOUS ALGEBRAICEQUATIONS (LINEAR) USING ITERATIVE METHODS, NUMERICAL DIFFERENTIATION ANDINTEGRATION NUMERICAL SOLUTION OF 1ST AND 2^{ND} ORDER DIFFERENTIAL EQUATIONS : (15 LECTURES)**
3.1	**Solution of simultaneous algebraic equations (linear) using iterative methods:** Gauss-Jordan Method, Gauss-Seidel Method.

3.2	**Numerical differentiation and Integration:** Numberical differentiation, Numerical integration using Trapezoidal Rule, Simpson's $1/3^{rd}$ and $3/8^{th}$ rules.
3.3	**Numerical solution of 1st and 2nd order differential equations:** Taylor series, Euler's Method, Modified Euler's Method, Runge-Kutta Method for 1^{st} and 2^{nd} Order Differential Equations.
UNIT 4	**LEAST-SQUARES REGRESSION LINEAR PROGRAMMING : (15 LECTURES)**
4.1	**Least-Squares Regression:** Linear Regression, Polynomial Regression, Multiple Linear Regression, General Linear Least Squares, Nonlinear Regression
4.2	**Linear Programming:** Linear optimization problem, Formulation and Graphical solution, Basic solution and Feasible solution
UNIT 5	**RANDOM VARIABLES DISTRIBUTIONS: (15 LECTURES)**
5.1	**Random variables:** Discrete and Continuous random variables, Probability density function, Probability distribution of random variables, Expected value, Variance.
5.2	**Distributions:** Discrete distributions: Uniform, Binomial, Poisson, Bernoulli, Continuous distributions: uniform distributions, exponential, (derivation of mean and variance only and state other properties and discuss their applications) Normal distribution state all the properties and its applications.

REFERENCES:

- Introductory methods of numerical analysis – 5th Edition by Sastry, S. S.
- Fundamental of Mathematical statistics – 11th Revised Edition by Gupta, S.C &Kapoor, V.K
- Introduction to Operations Research - 10th Edition by Hillier, Frederick, and others.

NAME OF THE COURSE	NUMERICAL STATISTICAL METHODS PRACTICAL	
CLASS	FYBSCIT	
COURSE CODE	SBTTECP204	
NUMBER OF CREDITS	2	
NUMBER OF LECTURES PER WEEK	3	
TOTAL NUMBER OF LECTURES PER SEMESTER	45	
EVALUATION METHOD	INTERNAL ASSESSMENT	SEMESTER END EXAMINATION
TOTAL MARKS	---	50
PASSING MARKS	---	20

List of Practical	
1.	**Iterative Calculation**
1.1	Program for iterative calculation.
1.2	Program to calculate the roots of a quadratic equation using the formula.
1.3	Program to evaluate e^x using infinite series.
2.	**Solution of algebraic and transcendental equations:**
2.1	Program to solve algebraic and transcendental equation by bisection method.
2.2	Program to solve algebraic and transcendental equation by false position method.
2.3	Program to solve algebraic and transcendental equation by Secant method.
2.4	Program to solve algebraic and transcendental equation by Newton Raphson method.
3.	**Interpolation**
3.1	Program for Newton's forward interpolation.
3.2	Program for Newton's backward interpolation.
3.3	Program for Lagrange's interpolation.
4.	**Solving linear system of equations by iterative methods**
4.1	Program for solving linear system of equations using Gauss Jordan method.
4.2	Program for solving linear system of equations using Gauss Seidel method.
5.	**Numerical Differentiation**
5.1.	Programing to obtain derivatives numerically.

6.	**Numerical Integration**
6.1	Program for numerical integration using Trapezoidal rule.
6.2	Program for numerical integration using Simpson's $1/3^{rd}$ rule.
6.3	Program for numerical integration using Simpson's $3/8^{th}$ rule.
7.	**Solution of differential equations**
7.1	Program to solve differential equation using Euler's method
7.2	Program to solve differential equation using modified Euler's method.
7.3	Program to solve differential equation using Runge-kutta 2^{nd} order and 4^{th} order methods.
8.	**Regression**
8.1	Program for Linear regression.
8.2	Program for Polynomial Regression.
8.3	Program for multiple linear regression.
8.4	Program for non-linear regression.
9.	**Random variables and distributions**
9.1	Program to generate random variables.
9.2	Program to fit binomial distribution.
9.3	Program to fit Poisson distribution.
10.	**Distributions**
10.1	Program for Uniform distribution.
10.2	Program for Bernoulli distribution
10.3	Program for Negative binomial distribution.

Semester – II		
NAMEOF THE COURSE	GREEN COMPUTING	
CLASS	FYBSc IT	
COURSE CODE	SBTTEC205	
NUMBER OF CREDITS	2	
NUMBER OF LECTURES PER WEEK	5	
TOTAL NUMBR OF LECTURES PER SEMESTER	75	
EVALUATION METHOD	INTERNAL ASSESSMENT	SEMESER END EXAMINATION
TOTAL MARKS	25	75
PASSING MARKS	10	30

COURSE OBJECTIVES:

CO 1.	The goal of studying green computing is to attain economic viability and improve the way computing devices are used.
CO 2.	Reduce the use of hazardous materials, maximize energy efficiency during the product&s lifetime
CO 3.	Students learn how to measure computer power usage, minimize power usage, procure sustainable hardware, design green data centers, recycle computer equipment, configure computers to minimize power, use virtualization to reduce the number of servers, and other green technologies.
CO 4.	Students learn to examine cooling issues in the datacenter as well as where you can save money, and it provides some tips for adding cooling capacity without spending more money than need to
CO 5.	Students learn different metrics to track and analyze greening of information systems

COURSE LEARNING OUTCOMES:

CLO 1.	Listing organizations environmental issues and explain how to measure its carbon footprint.
CLO 2.	To minimize power usage and maximize cooling needs.
CLO 3.	To change to change the way we work and to create a paperless environment
CLO 4.	To identify recycling methods and hardware considerations
CLO 5.	To improve the Technology Infrastructure and analyze the Organizational Check-ups

UNIT 1	**OVERVIEW AND ISSUES INITIATIVES AND STANDARDS : (15 LECTURES)**
1.1	**Overview and Issues:** Problems: Toxins, Power Consumption, Equipment Disposal, Company's Carbon Footprint: Measuring, Details, reasons to bother, Plan for the Future, Cost Savings: Hardware, Power.
1.2	**Initiatives and Standards:** Global Initiatives: United Nations, Basel Action Network, Basel Convention, North America: The United States, Canada, Australia, Europe, WEEE Directive, RoHS, National Adoption, Asia: Japan, China, Korea.
UNIT 2	**MINIMIZING POWER USAGE COOLING : (15 LECTURES)**
2.1	**Minimizing Power Usage:** Power Problems, Monitoring Power Usage, Servers, Low-Cost Options, Reducing Power Use, Data De-Duplication, Virtualization, Management, Bigger Drives, Involving the Utility Company, LowPower Computers, PCs, Linux, Components, Servers, Computer Settings, Storage, Monitors, Power Supplies, Wireless Devices, Software.

2.2	**Cooling:** Cooling Costs, Power Cost, Causes of Cost, Calculating Cooling Needs, Reducing Cooling Costs, Economizers, On-Demand Cooling, HP's Solution, Optimizing Airflow, Hot Aisle/Cold Aisle, Raised Floors, CableManagement, Vapour Seal, Prevent Recirculation of Equipment Exhaust, Supply Air Directly to Heat Sources, Fans, Humidity, Adding Cooling, Fluid Considerations, System Design, Datacentre Design, Centralized Control, Design for Your Needs, Put Everything Together.
UNIT 3	**CHANGING THE WAY OF WORK GOING PAPERLESS: (15 LECTURES)**
3.1	**Changing the Way of Work:** Old Behaviours, starting at the Top, Process Reengineering with Green in Mind, Analysing the Global Impact of Local Actions, Steps: Water, Recycling, Energy, Pollutants, Teleworkers and Outsourcing, Telecommuting, Outsourcing, how to Outsource.
3.2	**Going Paperless:** Paper Problems, The Environment, Costs: Paper and Office, Practicality, Storage, Destruction, Going Paperless, Organizational Realities, Changing Over, Paperless Billing, Handheld Computers vs. the Clipboard, Unified Communications, Intranets, What to Include, Building an Intranet, Microsoft OfficeSharePoint Server 2007, Electronic Data Interchange (EDI), Nuts and Bolts, Value Added Networks, Advantages, Obstacles.
UNIT 4	**RECYCLING HARDWARE CONSIDERATIONS :(15 LECTURES)**
4.1	**Recycling:** Problems, China, Africa, Materials, Means of Disposal, Recycling, Refurbishing, Make the Decision, Life Cycle, from beginning to end, Life, Cost, Green Design, Recycling Companies, Finding the Best One, Checklist, Certifications, Hard Drive Recycling, Consequences, cleaning a Hard Drive, Pros and cons of each method, CDs and DVDs, good and bad about CD and DVDs disposal, Change the mind-set, David vs.America Online
4.2	**Hardware Considerations:**

	Certification Programs, EPEAT, RoHS, Energy Star, Computers, Monitors, Printers,Scanners, All-in-Ones, Thin Clients, Servers, Blade Servers, Consolidation, Products, Hardware Considerations, Planned Obsolescence, Packaging, Toxins, Other Factors, Remote Desktop, Using Remote Desktop, Establishing a Connection, In Practice
UNIT 5	**GREENING YOUR INFORMATION SYSTEMS : (15 LECTURES)**
5.1	**Greening Your Information Systems:** Initial Improvement Calculations, Selecting Metrics, Tracking Progress, Change Business Processes, Customer Interaction, Paper Reduction, Green Supply Chain, Improve Technology Infrastructure, Reduce PCs and Servers, Shared Services, Hardware Costs, Cooling.
5.2	**Staying Green:** Organizational Check-ups, Chief Green Officer, Evolution, Sell the CEO, SMART Goals, Equipment Check-ups, Gather Data, Tracking the data, Baseline Data, Benchmarking, Analyse Data, Conduct Audits, Certifications, Benefits, Realities, Helpful Organizations.

REFERENCES:

- Green IT Toby Velte,Anthony Velte,Robert Elsenpeter, McGraw Hill 2008
- Green Computing Tools and Techniques for Saving Energy, Money and Resources, Bud E.Smith CRC Press 2014
- Green IT Deepak Shikarpur, Vishwkarma Publications, 2014

NAME OF THE COURSE	GREEN COMPUTING PRACTICAL	
CLASS	FYBSCIT	
COURSE CODE	SBTTECP205	
NUMBER OF CREDITS	2	
NUMBER OF LECTURES PER WEEK	3	
TOTAL NUMBER OF LECTURES PER SEMESTER	45	
EVALUATION METHOD	INTERNAL ASSESSMENT	SEMESTER END EXAMINATION
TOTAL MARKS	---	50
PASSING MARKS	---	20

Project and Viva Voce	
1.	A project should be done based on the objectives of Green Computing. A report of minimum 50 pages should be prepared. The report should have a font size of 12, Times new roman and 1.5 line spacing. The headings should have font size 14. The report should be hard bound.
2.	The project can be done individually or a group of two students.
3.	The students will have to present the project during the examination.
4.	A certified copy of the project report is essential to appear for the examination.

ASSESSMENT DETAILS:(this will be same for all the theory papers)

Internal Assessment (25 marks)

Part 1: Project Work (**20 Marks**) / Test

- At the beginning of the semester, students should be assigned projecttopics drawn from Unit 1 to Unit 5.
- Students can work in groups of not more than 3 per topic.
- Project Marks will be divided as written submission: 10 Marks &Presentation & Viva: 10 marks)
- The Project/Assignment can take the form of Street-Plays/Power-Point Presentations/Poster Exhibitions and similar other modes of presentation appropriate to the topic.
- Students must submit a hard copy of the Project before the lastteaching day of the semester.

Part 2: Attendance **– 05marks**

Semester End Examination – External Assessment (75 marks)

- The duration of the paper will be two and a half hours.
- There shall be five compulsory questions
- Q1-5 shall correspond to the five units. Q1-5 shall contain an internal choice (attempt any3 of 6). Q1-5 shall carry a maximum of 15 marks

Practical Assessment (for papers with practicals)

- The duration of the practical exam will be two and a half hours.
- The students are allowed to write the paper if the attendance for practicals is more than 75%
- To appear in the practical exam, students must bring a properly certified journal.

Evaluation Scheme:

1. **Internal Evaluation (25 Marks).**
   i. **Test: 1 Class test of 20 marks. (Can be taken online)**

Q	**Attempt *<u>any four</u>* of the following:**	**20**
a.		
b.		
c.		
d.		
e.		
f.		

ii. **5 marks: Active participation in the class, overall conduct, attendance.**

iii.

2. **External Examination: (75 marks)**

	All questions are compulsory	
Q1	**(Based on Unit 1) Attempt *any three* of the following:**	**15**
a.		
b.		
c.		
d.		
e.		
f.		
Q2	**(Based on Unit 2) Attempt *any three* of the following:**	**15**
Q3	**(Based on Unit 3) Attempt *any three* of the following:**	**15**
Q4	**(Based on Unit 4) Attempt *any three* of the following:**	**15**
Q5	**(Based on Unit 5) Attempt *any three* of the following:**	**15**

3. **Practical Exam: 50 marks**

A Certified copy journal is essential to appear for the practical examination.

1.	**Practical Question 1**	**20**
2.	**Practical Question 2**	**20**
3.	**Journal**	**5**
4.	**Viva Voce**	**5**

OR

1.	**Practical Question**	**40**
2.	**Journal**	**5**
3.	**Viva Voce**	**5**