SIES COLLEGE OF COMMERCE & ECONOMICS
AUTONOMOUS
DEPARTMENT OF INFORMATION TECHNOLOGY

Sr. No.	Heading	Particulars
1	Title of the course	B. Sc. (Information Technology)
2	Eligibility for admission	HSC or Equivalent with Mathematics as Compulsory Subject
3	Minimum percentage	45 %
4	Semesters	I & II
5	Level	UG
6	Pattern	03 years & 06 semesters CBGS
7	To be implemented from	From Academic year 2020-21 in a progressive manner

(WITH EFFECT FROM THE ACADEMIC YEAR 2020-2021)

Semester I			
Course Code	**Course Type**	**Course Title**	**Credits**
BITS101	Core Subject	Imperative Programming	2
BITS102	Core Subject	Digital Electronics	2
BITS103	Core Subject	Operating Systems	2
BITS104	Core Subject	Discrete Mathematics	2
BITS105	Ability Enhancement Skill Course	Communication Skills	2
BITS1P1	Core Subject Practical	Imperative Programming Practical	2
BITS1P2	Core Subject Practical	Digital Electronics Practical	2
BITS1P3	Core Subject Practical	Operating Systems Practical	2
BITS1P4	Core Subject Practical	Discrete Mathematics Practical	2
BITS1P5	Ability Enhancement Skill Course Practical	Communication Skills Practical	2
		Total Credits	**20**

Semester II			
Course Code	**Course Type**	**Course Title**	**Credits**
BITS201	Core Subject	Object oriented Programming	2
BITS202	Core Subject	Microprocessor Architecture	2
BITS203	Core Subject	Web Programming	2
BITS204	Core Subject	Numerical and Statistical Methods	2
BITS205	Ability Enhancement Skill Course	Green Computing	2
BITS2P1	Core Subject Practical	Object Oriented Programming Practical	2
BITS2P2	Core Subject Practical	Microprocessor Architecture Practical	2
BITS2P3	Core Subject Practical	Web Programming Practical	2
BITS2P4	Core Subject Practical	Numerical and Statistical Methods Practical	2
BITS2P5	Ability Enhancement Skill Course Practical	Green Computing Practical	2
		Total Credits	**20**

SEMESTER I

Imperative Programming

COURSE CODE: BITS101 **COURSE CREDIT: 02**

Course Objectives:

- To introduce different programming paradigms and develop logic for writing high level language programs.
- To familiarize the students with the basic understanding of flowcharts and algorithms.

Sr. No	Modules/Units	No of Lectures
1.	**Introduction:** Types of Programming languages, History, features and application. Simple program logic, program development cycle, pseudo code statements and flowchart symbols, sentinel value to end a program, programming and user environments, evolution of programming models., desirable program characteristics. **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.	12
2.	**Operators and Expressions**: Arithmetic operators, unary operators, relational and logical operators, assignment operators, assignment operators, the conditional operator, library functions. **Data Input and output:** Single character input and output, entering input data, scanf function, printf function, gets and puts functions, interactive programming.	12
3.	**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 . **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.	12
4.	**Program structure:** Storage classes, automatic variables, external variables, static variables, multifile programs, more library functions, **Preprocessor:** Features, #define, #undef, Once-Only Include Files and #include, Directives and Macros , types of macros **Arrays:** Definition, processing, passing arrays to functions, multidimensional arrays, arrays and strings.	
	Pointers: Fundamentals, declarations, Pointers Address Operators, Pointer Type Declaration, Pointer Assignment, Pointer Initialization, Pointer Arithmetic, Functions and Pointers, Arrays And Pointers,	

5.	Pointer Arrays, passing functions to other functions **File Handling:** Why files are needed? , File Input/Output, Data Organization, File Operations, Text Files and Binary Files, Working with files. **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 **Advanced Industrial Programming** : Logic & Functionality of Programming in Industries, Service Based and Product Based IT Industries, Autonomous problem analysis/solution, Use of Available Resources, Developing a Software through C, Importance of reusable and maintainable code, Updation of Programming Languages in IT industry	12

REFERENCE BOOKS:

1. Programming with C Byron Gottfried,Tata McGRAW-Hill, 2nd edition,1996
2. Programming Logic and Design, Joyce Farell, Cengage Learning, 8th edition, 2014
3. "C" Programming", Brian W. Kernighan and Denis M.Ritchie., PHI, 2nd edition
4. Let us C, Yashwant P.Kanetkar, BPB publication
5. C for beginners, Madhusudan Mothe, X-Team Series, 1st edition, 2008
6. 21st Century C Ben Klemens, OReilly, 1st edition, 2012

Imperative Programming Practical

COURSE CODE: BITS1P1 **COURSE CREDIT: 02**

Course Objectives:

- To make students develop algorithms and write C programs for the same.

List of Practicals:	
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 and 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 programs 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 $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$
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 using pointers 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.	**Pointers**
a.	Write a program to demonstrate the use of pointers.
b.	Write a program to perform addition and subtraction of two pointer variables.
10.	**Structures and Unions**
a.	Programs on structures.
b.	Programs on unions.
11.	**File Handling in C**
a.	C program to read name and marks of n number of students and store them in a file.
b.	Write to a binary file using fwrite()
c.	Read from a binary file using fread()

Digital Electronics

COURSE CODE: BITS102 **COURSE CREDIT: 02**

Course Objectives:

- To make students learn different types of number systems.
- To make students acquire the basic knowledge of digital logic levels and application of knowledge to understand digital electronics circuits.
- To prepare students to perform the analysis and design of various digital electronic circuits.

Sr. No	Modules/Units	No of Lectures
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, 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
2.	**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 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

3.	**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, Comparator.	12
4.	**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 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.	12
5.	**Counters:** Introduction, Asynchronous counter, Terms related to counters, IC7493 (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. **Introduction To Digital Logic Families:** Resistor-Transistor logic (RTL), Diode-Transistor Logic (DCL), Transistor-Transistor Logic (TTL), Emitter Coupled Logic (ECL), MOS Logic, CMOS Logic	12

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

Digital Electronics Practical

COURSE CODE: BITS1P2 **COURSE CREDIT: 02**

Course Objectives:

- To acquire the basic knowledge of digital logic levels and application of knowledge to understand digital electronics circuits.
- To prepare students to perform the analysis and design of various digital electronic circuits.

List of Practical	
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.

Operating Systems

COURSE CODE: BITS103 **COURSE CREDIT: 02**

Course Objectives:

- To make students learn the fundamentals of Operating Systems.
- To make students learn the mechanisms of OS to handle processes and threads and their communication.
- To make students gain knowledge on distributed operating system concepts that includes architecture, mutual exclusion algorithms, deadlock detection algorithms and agreement protocols.

Sr. No	Modules/Units	No of Lectures
1.	**Introduction:** What is an operating system? History of operating system, computer hardware, different operating systems, operating system concepts, system calls, operating system structure.	12
2.	**Processes and Threads:** Processes, threads, interprocess communication, scheduling, IPC problems. **Memory Management:** No memory abstraction, memory abstraction: address spaces, virtual memory, page replacement algorithms, design issues for paging systems, implementation issues, segmentation.	12
3.	**File Systems:** Files, directories, file system implementation, file- system management and optimization, MS-DOS file system, UNIX V7 file system, CD ROM file system **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.	12
4.	**Deadlocks:** Resources, introduction to deadlocks, the ostrich algorithm, deadlock detection and recovery, deadlock avoidance, deadlock prevention, issues **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.	12
5.	**Multiple Processor Systems:** Multiprocessors, Multicomputers, distributed systems. **Security:** The security environment: threats, attackers. Formal models of Secure Systems: Multilevel Security, Covert Channels. Basics of Cryptography: Secret Key Cryptography, Public-Key Cryptography, Digital Signatures.	12

REFERENCE BOOKS:

1. Modern Operating Systems, Andrew S. Tanenbaum, Herbert Bos, Pearson, 4th, 2014
2. Operating Systems – Internals and Design Principles, Willaim Stallings, Pearson, 8th, 2009
3. Operating Systems, Godbole and Kahate, McGraw Hill, 3rd

Operating Systems Practical

COURSE CODE: BITS1P3 **COURSE CREDIT: 02**

Course Objectives:

- To make students learn different DOS and Linux commands using command line interface and shell.
- To make students understand different utilities provided by Windows and Linux operating system

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:**
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**
11.	Case Study on LINUX, WINDOWS & ANDROID Operating Systems.

Discrete Mathematics

COURSE CODE: BITS104 **COURSE CREDIT: 02**

Course Objectives:

- To make students learn the basic principles of set, basic set equalities, the basic concepts of relations and functions and the basic concepts of graphs and trees.
- To make students learn writing an argument using logical notation and determine if the argument is valid or invalid.
- To make students learn the basic concepts of data structures in mathematics.

Sr. No	Modules/Units	No of Lectures
1.	**Introduction**: Variables, The Language of Sets, The Language of Relations and Function. **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.	12
2.	**Quantified Statements**: Predicates and Quantified Statements, Statements with Multiple Quantifiers, Arguments with Quantified Statements. **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
3.	**Sequences, Mathematical Induction, and Recursion**: Sequences, Mathematical Induction, Strong Mathematical Induction and the Well-Ordering 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. **Functions**: Functions Defined on General Sets, One-to-One and Onto, Inverse Functions, Composition of Functions, Cardinality with Applications to Computability.	12
4.	**Relations**: Relations on Sets, Reflexivity, Symmetry, and Transitivity, Equivalence Relations, Partial Order Relations, Hasse Diagram, Lattice as Poset, Properties of Lattices **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

5.	**Counting and Probability**: Introduction, Possibility Trees and the Multiplication Rule, 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, Probability Axioms and Expected Value, Conditional Probability, Bayes‟ Formula, and Independent Events.	12

REFERENCE BOOKS:

1. Discrete Mathematics with Applications, Sussana S. Epp, Cengage Learning .
2. Discrete Mathematics, Schaum‟s Outlines Series, Seymour Lipschutz, Marc Lipson, Tata Mc Graw Hill
3. Discrete Mathematics and its Applications, Kenneth H. Rosen, Tata Mc Graw Hill 10. Elements of Discrete Mathematics, C. L. Liu, Tata McGraw-Hill

Discrete Mathematics Practical

COURSE CODE: BITS1P4 **COURSE CREDIT: 02**

Course Objectives:

- To make students learn the main features of SCILAB program development environment.
- To make students learn to implement simple mathematical functions in SCILAB.

List of Practical: 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 Evaluation
d.	Greatest Common Divisor
3.	**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
4.	**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
5.	**Graph Theory**
a.	Paths and Connectivity
b.	Minimum Spanning Tree
c.	Isomorphism
6.	**Directed Graphs**
a.	Adjacency Matrix
b.	Path Matrix
7.	**Properties of Integers**
a.	Division Algorithm
b.	Primes
c.	Euclidean Algorithm
d.	Fundamental Theorem of Arithmetic

e.	Congruence Relation
f.	Linear Congruence Equation
8.	**Algebraic Systems**
a.	Properties of Operations
b.	Roots of Polynomials
9.	**Boolean Algebra**
a.	Basic Definitions in Boolean Algebra
b.	Boolean Algebra as Lattices
10.	**Recurrence Relations**
a.	Linear Homogeneous Recurrence Relations with Constant Coefficients
b.	Solving Linear Homogeneous Recurrence Relations with Constant Coefficients
c.	Solving General Homogeneous Linear Recurrence Relations

Communication Skills

COURSE CODE: BITS105 **COURSE CREDIT: 02**

Course Objectives:

- To develop awareness of the complexity of the communication process.
- To develop effective listening skills in students so as to enable them to comprehend instructions and become a critical listener.
- To develop effective oral skills so as to enable students to speak confidently interpersonally as well as in large groups.
- To develop effective writing skills so as enable students to write in a clear, concise, persuasive and audience centred manner.
- To develop ability to communicate effectively with the help of electronic media.

Sr. No	Modules/Units	No of Lectures
1	**Understanding Business Communication** A) Nature and Scope of Communication. B)**Methods & Effective Communication** Verbal Communication, The 7 Cs of Communication, Non-verbal Communication, C)Technology Enabled Communication D) Barriers to Communication(Physical, Linguistic, Psychological & Cross-Cultural Barriers)	14
2	**Writing Business Messages and Documents:** A)Business writing and correspondence, Career building and Resume writing B**)** Business Reports & Proposals C) Instructions and Summary writing	12
3	**Developing Oral Communication Skills for Business:** A) Effective Listening & Reading, B)Business Presentations, Public Speaking, Interviews C) Meetings, Conferences, Group Discussions and Team Presentations.	14
4	**Understanding Specific Communication Needs:** A) Corporate Communication(Channels-Formal and Informal & PR) B) Ethics in Business Communication C) Etiquette in Business Communication	12
5	**Presentation Process:** Planning and Executing the presentations, Impressing the Audience by performing. Planning stage- Brain storming, Mind maps. Executing stage: Chunking theory, creating outlines, Use of templates, Adding graphics, Use of font, colour, layout. Importance of Practice and Performance.	08
	TOTAL	**60**

References:

1. Banerjee, Bani P (2005) Foundation of Ethics in Mangement Excel Books
2. Bhargava and Bhargava91971) Company Notices, Meetings and Regulations
3. Black, Sam (1972) Practical Public Relations, E.L.B.S. London.
4. Bovee Courtland,L and Thrill, John V(1989) Business Communication, Today McGraw Hill, New York, Taxman Publication.
5. Fisher Dalmar, (1999), Communication in Organisation, Jaico Pub House, Mumbai, Delhi.
6. Frailley, L.E. (1982) Handbook of Business Letters, Revised Edn. Prentice Hall Inc. .
7. Garlside, L.E. (1980) Modern Business Correspondence, McDonald and Evans Ltd. Plymouth.
8. Lesikar, Raymond V and Petit, John D.(1994) Business Communication: Theory and Application ,Richard D. Irwin Inc. Ilinois.

Communication Skills Practical

COURSE CODE: BITS1P5 **COURSE CREDIT: 02**

Practical work:

1. Identifying two way communication in given situations
2. Identifying body language in given visual situations
3. Exercises with instructions
4. Resume writing
5. Mock Interviews
6. Summary writing
7. Use of Word processing tools, Spread sheet tools and Presentation tools for communication
8. Individual and team presentations

SEMESTER II

Object Oriented Programming

COURSE CODE: BITS201 **COURSE CREDIT: 02**

Course Objectives:

- To enable students to understand object oriented programming.
- To explain the difference between object oriented programming and procedural programming.
- To teach the various types of statements and looping constructs.

Sr. No	Modules/Units	No of Lectures
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. **Principles of OOPS:** OOPS Paradigm, Basic Concepts of OOPS: Objects, Classes, Data Abstraction and Data Encapsulation, Inheritance, Polymorphism, Dynamic Binding, Message Passing	12
2.	**Classes and Objects:** Simple classes (Class specification, class members accessing), The Main function, Function prototyping, Defining member functions, passing object as an argument, Returning object from functions, friend classes, Pointer to object, Array of pointer to object. **Constructors and Destructors:** Introduction, Default Constructor, Parameterized Constructor and examples, Destructors	12
3.	**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, **Virtual Functions**: Introduction and need, Pure Virtual Functions, Static Functions, this Pointer, abstract classes, virtual destructors.	12
4.	**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. **Exception Handling:** Introduction, Exception Handling Mechanism, Concept of throw & catch with example	12
5.	**Templates:** Introduction, Function Template and examples, Class Template and examples. **Working with Files:** Introduction, File Operations, Various File Modes, File Pointer and their Manipulation **Debugging:** Enabling all warnings and error messages, Insisting on clean compiles, Using a clear and consistent coding style, Limiting the visibility, Adding comments to your code while you write it, Single-stepping every path at	12

	least once, Avoiding overloaded operators, Heap handling, Using exceptions to handle errors, Avoiding multiple inheritance.	

REFERENCE BOOKS:

1. C++ for DUMMIES , Stephen Randy Davis , Wiley Publishing, Inc.
2. Object Oriented Programming with C++, E. Balagurusamy , Tata McGraw Hill
3. Object Oriented Analysis and Design, Object Oriented Analysis and Design, Timothy Budd TMH
4. Mastering C++ , K R Venugopal, Rajkumar Buyya, T Ravishankar, Tata McGraw Hill

Object Oriented Programming Practical

COURSE CODE: BITS2P1 **COURSE CREDIT: 02**

Course Objectives:

- To enable students to program using C++ features such as operator overloading, inheritance, polymorphism, file I/O, exception handling, etc.
- To enable students to build C++ classes using appropriate encapsulation and design principles
- Ultimate goal is to make students a good programmer.

List of Practical: To be implemented using Object Oriented Language	
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.Where readNo() 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 inheritance.
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
c.	Show the implementation of exception handling for using the pointers.
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.	Show the implementation for the following
b.	Show the implementation of template class library for swap function.
c.	Design the template class library for sorting ascending to descending and vice versa

Microprocessor Architecture

COURSE CODE: BITS202 **COURSE CREDIT: 02**

Course Objectives:

- To enable the students to learn the concept of assembly languages and acquire knowledge about 8085 microprocessor.
- To educate the students about 8085 architecture and instruction set.

Sr. No	Modules/Units	No of Lectures
1.	**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. **8085 Microprocessor Architecture and Memory Interface:** Introduction, 8085 Microprocessor unit, 8085-Based Microcomputer, 8085 Machine Cycles & Bus Timings, 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.	12
2.	**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. **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. **Introduction to 8085 Instructions:** Data Transfer Operations, Arithmetic Operations, Logic Operation, Branch Operation, Writing Assembly Languages Programs, Debugging a Program.	12

3.	**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. Stacks and Sub-Routines: Stack, Subroutine, Restart, Conditional Call, Return Instructions, Advanced Subroutine concepts.	**12**
4.	**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. 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
5.	**The Pentium and Pentium Pro microprocessors**: Introduction, Special Pentium registers, Memory management, Pentium instructions, Pentium Pro microprocessor, Special Pentium Pro features. **Core 2 and later Microprocessors**: Introduction, Pentium II software changes, Pentium IV and Core 2, i3, i5 and i7. SUN SPARC Microprocessor: Architecture, Register file, data types and instruction format	12

REFERENCE BOOKS:

1. Microprocessors Architecture, Programming and Applications with the 8085, Ramesh Gaonkar, PENRAM, Fifth, 2012
2. Computer System Architecture, M. Morris Mano,PHI
3. Structured Computer Organization, Andrew C. Tanenbaum, PHI

Microprocessor Architecture Practical

COURSE CODE: BITS2P2 **COURSE CREDIT: 02**

Course Objectives:

- To enable the students to learn the concept of assembly languages and acquire knowledge about 8085 microprocessor.
- To educate the students about 8085 architecture and instruction set.

List of Practical	
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 numbersstored 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. b. Consider the sum to be 16 bit number. Store the sum at memory locations4300H and 4301H
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. b. Consider the sum to be 16 bit number. Store the sum at memory locations4300H 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.	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.
d.	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
e.	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 ofthe list is 2000H. If byte is not found store 00 at 2200H and 2201H
f.	Two decimal numbers six digits each, are stored in BCD package form. Each number occupies 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
g.	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.	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.
c.	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.
d.	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
e.	Write an assembly language program to generate fibonacci number.
f.	Program to calculate the factorial of a number between 0 to 8.

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.	Multiplythe 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.	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
e.	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.
e.	Write a simple program to Split a HEX data into two nibbles and store it in memory
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

Web Programming

COURSE CODE: BITS203 **COURSE CREDIT: 02**

Course Objectives:

- To make the students learn website organization, HTML, graphics use, page and site design, with a brief look at CSS, and JavaScript.
- To enable students learn the basic and advanced PHP programming with Database connectivity using MYSQL.

Sr. No	Modules/Units	No of Lectures
1.	**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. **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.	12
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. **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	12

3.	**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)** : 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 **XML :** Introduction to XML, Anatomy of an XML document, Creating XML Documents, Creating XML DTDs, XML Schemas, XSL	12
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	12
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	12

REFERENCE BOOKS:

1. Web Design The Complete Reference, Thomas Powell, Tata McGraw Hill
2. HTML5 Step by Step, Faithe Wempen, Microsoft Press 2011
3. PHP 5.1 for Beginners, Ivan Bayross Sharanam Shah, SPD 2013
4. PHP Project for Beginners, SharanamShah, Vaishali Shah, SPD 2015
5. PHP 6 and MySQL Bible, Steve Suehring, Tim Converse, Joyce Park, Wiley 2009
6. JavaScript 2.0: The Complete Reference, Thomas Powell and Fritz Schneider, Tata McGraw Hill 2 nd
7. HTML and XHTML The Complete Reference, Thomas Powell, Tata McGraw Hill

Web Programming Practical

COURSE CODE: BITS2P3 **COURSE CREDIT: 02**

Course Objectives:

- To teach how to design websites and webpages using HTML, CSS, JavaScript and PHP. ·
- To be able to build dynamic webpages with back end connectivity using PHP and MYSQL.

Practical No	Details
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 with a form that uses all types of controls.
c.	Design a web page demonstrating different semantics
d.	Design a web page with different tables. Design a webpages using table so that the content appears well placed.
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.	**XML**
a.	Design a DTD, corresponding XML document and display it in browser using CSS.
b.	Design an XML document and display it in browser using XSL.
c.	Design XML Schema and corresponding XML document.
6.	**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.
7.	**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 1 0 1 0
8.	**String Functions and arrays**
a.	Write a PHP program to demonstrate different string functions.
b.	Write a PHP program to create one dimensional array.

9.	**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
10.	**Email**
a.	Write a program to send email with attachment.
11.	**Sessions and Cookies**
a.	Write a program to demonstrate use of sessions and cookies.

Numerical and Statistical Methods

COURSE CODE: BITS204 **COURSE CREDIT: 02**

Course Objectives:

- To make the students analyze the errors obtained in the numerical solution of problems. ·
- To help students to learn the use of appropriate numerical method to determine approximate solution of algebraic and transcendental equations, system of linear equations, ordinary differential equation and integration.
- To make students learn modeling and solving linear programming problems. · To make students learn an illustration and formulation of probability distribution and density functions.

Sr. No	Modules/Units	No of Lectures
1.	**Mathematical Modeling and Engineering Problem Solving**: A Simple Mathematical Model, Conservation Laws and Engineering Problems. **Approximations and Round-Off Errors**: Significant Figures, Accuracy and Precision, Error Definitions, Round-Off Errors. **Truncation Errors and the Taylor Series**: The Taylor Series, Error Propagation, Total Numerical Errors, Formulation Errors and Data Uncertainty	12
2.	**Module-2:** **Solutions of Algebraic and Transcendental Equations**: The Bisection Method, The Newton-Raphson Method, The Regula falsi method, The Secant Method. **Interpolation**: Forward Difference, Backward Difference, Newton‟s Forward Difference Interpolation, Newton‟s Backward Difference Interpolation, Lagrange‟s Interpolation.	12
3.	**Solution of simultaneous algebraic equations (linear) using iterative methods**: Gauss-Jordan Method, Gauss-Seidel Method. **Numerical differentiation and Integration**: Numerical differentiation, Numerical integration using Trapezoidal Rule, Simpson‟s 1/3rd and 3/8th rules. **Numerical solution of 1st and 2nd order differential equations**: Taylor series, Euler‟s Method, Modified Euler‟s Method, Runge Kutta Method for 1st and 2nd Order Differential Equations.	12

4.	**Least-Squares Regression**: Linear Regression, Polynomial Regression, Multiple Linear Regression, General Linear Least Squares, Nonlinear regression. **Linear Programming**: Linear optimization problem, Formulation and Graphical solution, Basic solution and Feasible solution	12
5.	**Random Variables**: Discrete and Continuous random variables, Probability density function, Probability distribution of random variables, Expected value, Variance, Moments. **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.	12

REFERENCE BOOKS:

1. Introductory Methods of Numerical Analysis, S. S. Shastri, PHI
2. Numerical Methods for Engineers, Steven C. Chapra, Raymond P. Canale, Tata MCGraw Hill
3. Fundamentals of Mathematical Statistics, S. C. Gupta, V. K. Kapoor, Sultan Chand & Sons
4. Numerical Analysis, Richard L. Burden, J. Douglas Faires, Cengage Learning

Numerical and Statistical Methods Practical

COURSE CODE: BITS2P4 **COURSE CREDIT: 02**

Course Objectives:

- · Students will be taught to write programs for various numerical and statistical methods.
-

List of Practical: Write the programs for the following using SCILAB	
1.	**Iterative Calculation**
a.	Program for iterative calculation.
b.	Program to calculate the roots of a quadratic equation using the formula.
c.	Program to evaluate ▯▯ using infinite series.
2.	**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.
3.	**Interpolation**
a.	Program for Newton‟s forward interpolation.
b.	Program for Newton‟s backward interpolation.
c.	Program for Lagrange‟s interpolation.
4.	**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.

5.	**Numerical Differentiation**
a.	Programing to obtain derivatives numerically.
6.	**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.
7.	**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.
8.	**Regression**
a.	Program for linear regression.
b.	Program for polynomial regression.
c.	Program for multiple linear regression.
d.	Program for non-linear regression.
9.	**Random Variables and Distributions:**
a.	Program to generate random variables.
b.	Program to fit binomial distribution.
c.	Program to fit Poisson distribution.

10.	**Distributions**
a.	Program for uniform distribution.
b.	Program for Bernoulli distribution
c.	Program for negative binomial distribution.

Green Computing

COURSE CODE: BITS205 **COURSE CREDIT: 02**

Course Objectives:

- To make students understand and develop special skills such as energy efficiency, ethical IT assets disposal, carbon footprint estimation, reporting and development of green products, applications and services.
- To educate students about appropriate hardware and software for feasible operations.

Sr. No	Modules/Units	No of Lectures
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. **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.	12
2.	**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. **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, Cable Management, 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.	12
3.	**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. **Going Paperless:** Paper Problems, The Environment, Costs: Paper and	12

	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 Office SharePoint Server 2007, Electronic Data Interchange (EDI), Nuts and Bolts, Value Added Networks, Advantages, Obstacles.	
4.	**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 **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	12
5.	**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. **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.	12

REFERENCE BOOKS:

1. Green IT , Toby Velte, Anthony Velte, Robert Elsenpeter, McGraw Hill

Green Computing Practical

COURSE CODE: BITS2P5 **COURSE CREDIT: 02**

Course Objectives:

- Students will be able to learn how different countries in world dispose e-waste and implement Green Computing practices.
- Students will develop awareness about e-waste disposal in surroundings.

Project and Viva Voce

1. Students will create awareness about e-waste impacts on environment and human health, collect e-waste and handover to e-waste recycling industry.

2. Case Studies : Dell, Hewlett-Packard, University of Wisconsin–River Falls, Wal-Mart

3. 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.

4. The project can be done individually or a group of two students.

5. The students will have to present the project during the examination.

6. A Certified copy of the project report is essential to appear for the examination.