**AC/14.06.21/RS1**

SIES

College of Arts,
Science &
Commerce

RISE WITH EDUCATION

**Sion (West), Mumbai – 400022.**

# (Autonomous)

## Faculty: Science

## Program: B.Sc.

## Subject: INFORMATION TECHNOLOGY

**Academic Year: 2021 – 2022**

# F.Y.B.Sc.

**Credit Based Semester and Grading Syllabi approved by Board of Studies in Information Technology to be brought into effect from June 2021.**

# PREAMBLE

With the growing dependence of computers in our lives, information technology has become an indispensable part of it. Information Technology is one of the most popular and fastest growing industries which provides millions of jobs to people across the globe. To address this demand for IT professionals, this three-year BSc programme in Information Technology is being introduced. The BSc(IT) Programme will enable the students to have sound knowledge in the core IT subjects through which they can develop user-friendly and scalable applications. This programme aims to prepare students for careers requiring IT expertise and to develop skills required to work in teams within organisations.

## PROGRAMME SPECIFIC OUTCOMES

| PSO No | Details |
|---|---|
| **PSO1** | **Sound Knowledge:**<br>Demonstrate the knowledge of core IT concepts and apply them to develop a user-friendly, scalable and robust applications |
| **PSO2** | **Critical and Rational Thinking:**<br>Exhibit higher order skills to adapt to the everchanging technological environment |
| **PSO3** | **Logic Building and Programming Skills:**<br>The ability to apply logic to problem solving and acquire proficiency in various programming languages. |
| **PSO4** | **Develop Communication skills and environmental awareness:**<br>Knowledge of basic communication skills and environmentally friendly computing practices. Work and communicate effectively in teams within organizations. |
| **PSO5** | **Enable Employability:**<br>Create computer experts, who can be directly employed or start his/her own work as Web Designer, Database User, Programmer, Testing professional, Designer of a System and Network implementer. |
| **PSO6** | **Pursue Higher Education:**<br>Make students competent to take up advanced degree courses like MCA, MSc(CS), MSc(IT) and MBA etc. |

# FYBSc(IT) - Semester I

| Course Code | Course Type | Course Title | Periods per week (1 Period = 50 min) | Credits |
|---|---|---|---|---|
| SIUSIT11 | Core Subject | Imperative Programming | **5** | **2** |
| SIUSIT12 | Core Subject | Digital Electronics | **5** | **2** |
| SIUSIT13 | Core Subject | Web Programming | **5** | **2** |
| SIUSIT14 | Core Subject | Discrete Mathematics | **5** | **2** |
| SIUSIT15 | Ability Enhancement Skill Course | Communication Skills | **5** | **2** |
| | | | | |
| SIUSITP11 | Core Subject Practical | Imperative Programming Practical | **3** | **2** |
| SIUSITP12 | Core Subject Practical | Digital Electronics Practical | **3** | **2** |
| SIUSITP13 | Core Subject Practical | Web Programming Practical | **3** | **2** |
| SIUSITP14 | Core Subject Practical | Discrete Mathematics Practical | **3** | **2** |
| SIUSITP15 | Ability Enhancement Skill Course Practical | Communication Skills Practical | **3** | **2** |
| **TOTAL CREDITS** | | | | **20** |

# FYBSc(IT) - Semester I

# Imperative Programming (SIUSIT11)

## Course Objective:

To acquaint learners about the importance of logical thinking in software development.

## Course Outcomes:

**Upon completion of this course, student will be able to:**

CO1: Write Decision making programs in C and explain the importance of switch statement.

CO2: Categorize the different data types in C, compare and contrast the various loop structures and examine the storage classes in C.

CO3: Inspect the built-in functions in C , compose user defined functions and use pointers to work with addresses in memory

CO4: Design programs using Arrays in C

CO5: Explain the use of standard library string functions and discuss the difference between structure and union in C.

| Unit | Contents | No. of Lectures |
|---|---|---|
| **I** | **Getting Started with C :** The C Character Set, Constants, Variables and Keywords, Types of C Constants, Rules for Constructing Integer Constants, Rules for Constructing Real Constants, Rules for Constructing Character Constants, Types of C Variables, Rules for Constructing Variable Names, C Keyword<br>**C Instructions :** Type Declaration Instruction, Arithmetic Instruction, Integer and Float Conversions, Type Conversion in Assignments, Hierarchy of Operations, Associativity of Operators<br>**The Decision Control Structure** : The if Statement, The if-else Statement, Nested if-else, Use of Logical Operators, The else if Clause, The ! Operator, Hierarchy of Operators Revisited, The Conditional Operators<br>**The Case Control Structure** : Decisions Using switch, switch Versus if-else Ladder, The goto Keyword | **12** |
| **II** | **Data Types Revisited**: Integers : long and short, signed and unsigned, Chars : signed and unsigned, Floats and Doubles<br>**The Loop Control Structure** : The while Loop, The for Loop, Nesting of Loops, For Loop, The Odd Loop, The break Statement, The continue Statement, The do-while Loop<br>**Storage Classes in C :** Automatic Storage Class, Register Storage Class, Static Storage Class, External Storage Class, Which to Use When | **12** |

| | | |
|---|---|---|
| **III** | **The C Preprocessor** : Features of C Preprocessor, Macro Expansion, Macros with Arguments, Macros versus Functions, File Inclusion<br>**Functions & Pointers**: What is a Function, Why Use Functions, Passing Values between Functions, Scope Rule of Functions, and Calling Convention<br>**Advanced Features of Functions** : Function Declaration and Prototypes, Call by Value ,Call by Reference, Recursion, An Introduction to Pointers, Pointer Notation, Function Calls using pointers | **12** |
| **IV** | **Arrays** : What are Arrays?, Array Initialization, Bounds Checking, Passing Array Elements to a Function, Pointers and Arrays, Passing an Entire Array to a Function, Two Dimensional Arrays, Initializing a 2-Dimensional Array, Memory Map of a 2-Dimensional Array | **12** |
| **V** | **Puppetting On Strings** : What are Strings, More about Strings, Pointers and Strings, Standard Library String Functions, strlen( ), strcpy( ), strcat( ), strcmp( ), Two-Dimensional Array of Characters<br>**Structures** : Why Use Structures, Declaring a Structure, Accessing Structure Elements, How Structure Elements are Stored, Array of Structures<br>**Unions :** Need for Union in C programming, Difference between structure and union, Syntax for Declaring a C union | **12** |

**Books and References**

| **Sr. No.** | **Title** | **Author/s** | **Publisher** | **Edition** | **Year** |
|---|---|---|---|---|---|
| **1** | Let us C | Yashwant P. Kanetkar | BPB | 16th | 2020 |
| **2** | Programming Logic and Design | Joyce Farell | Cengage Learning | 8th | 2014 |
| **3** | “C” Programming | Brian W. Kernighan and Denis M. Ritchie | PHI | 2nd | 2015 |
| **4** | C for beginners | Madhusudan | Mothe X-Team Series | 1st | 2008 |
| **5** | 21st Century C | Ben Klemens | OReilly | 1st | 2012 |

**Internal Evaluation: 40 Marks**

| **20 Marks** | **20 Marks** |
|---|---|
| Class Test | Projects to be developed in Scratch software |

# FYBSc(IT) - Semester I

## Imperative Programming Practical (SIUSITP11)

**List of Practical: (To be done in C Language)**

| | |
|---|---|
| **1.** | **Basic Programs:** |
| | a. Write a program to swap 2 numbers using 2 and 3 variables.<br>b. Write a program to find the area of rectangle, square and circle.<br>c. Write a program to find the volume of a cube, sphere, and cylinder. |
| **2.** | **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.<br>b. Write a program to check whether the number is even or odd.<br>c. Write a program to check whether the number is positive, negative or zero.<br>d. Write a program to find the factorial of a number.<br>e. Write a program to check whether the entered number is prime or not.<br>f. Write a program to find the largest of three numbers. |
| **3.** | **Conditional statements and loops(advanced)** |
| | a. Write a program to find the sum of squares of digits of a number.<br>b. Write a program to reverse the digits of an integer.<br>c. Write a program to find the sum of numbers from 1 to 100.<br>d. Write a programs to print the Fibonacci series.<br>e. Write a program to find the reverse of a number.<br>f. Write a program to find whether a given number is palindrome or not.<br>g. Write a program that solve the quadratic equation<br>h. Write a program to check whether the entered number is Armstrong or not.<br>i. Write a program to count the digit in a number |
| **4.** | **Programs on patterns:** |
| | a. Programs on different patterns. (any 2 patterns) |
| **5.** | **Functions:** |
| | a. Write a program to check whether a given number is prime number or not. Use user defined functions.<br>b. Write a program to find the sum of the digits of a number. Use user defined functions.<br>c. Write a program to swap 2 numbers. Use call by value and call by reference for the same. |
| **6.** | **Recursive functions** |
| | a. Write a program to find the factorial of a number using recursive function.<br>b. Write a program to find the sum of natural number using recursive function. |

| | |
|---|---|
| **7.** | **Arrays:** |
| | a. Write a program to find the largest value that is stored in the array.<br>b. Write a program using pointers to compute the sum of all elements stored in an array.<br>c. Write a program to arrange the ‘n’ numbers stored in the array in ascending and descending order.<br>d. Write a program that performs addition and subtraction of matrices.<br>e. Write a program that performs multiplication of matrices. |
| **8.** | **Pointers:** |
| | a. Write a program to swap 2 numbers using call by reference.<br>b. Write a program to find area and perimeter of a circle using call by reference. |
| **9.** | **Strings** |
| | a. Write a program to find the length of a string, concatenate 2 strings, copy a string from one location to the other using built-in and user defined functions. |
| **10** | **Structures and Unions** |
| | a. Write a program to accept 3 details of books and print the same using structures.<br>b. Accept 3 different primitive type of data and print the same. Use Union. |

# Digital Electronics (SIUSIT12)

## Course Objective:

To provide learners an insight into Digital Data Storage and the working of fundamental elements of digital electronics useful for digital communication.

## Course Outcomes:

**Upon completion of this course, student will be able to:**

CO1: Examine and apply the structure of various number systems, binary arithmetic and its applications in digital design

CO2: Apply the Boolean algebra using logic gates and Karnaugh Maps

CO3: Construct and design Combinational and Arithmetic circuits.

CO4: Construct and design Combinational and Sequential logic circuits.

CO5: Apply Sequential logic circuits to build Registers and Counters.

| Unit | Contents | No. of Lectures |
|---|---|---|
| I | **Number System:**<br>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.<br>**Binary Arithmetic:**<br>Binary addition, Binary subtraction, Negative number representation, Subtraction using 1's complement and 2's complement, Binary multiplication and division, Arithmetic in octal number system, Arithmetic in hexadecimal number system, BCD and Excess – 3 arithmetic. | 12 |
| II | **Boolean Algebra and Logic Gates:**<br>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.<br>**Minterm, Maxterm and Karnaugh Maps:**<br>Introduction, minterms and sum of minterm form, maxterm and Product of maxterm form, Reduction technique using Karnaugh maps – 2/3/4 variables K-maps, Grouping of variables in K-maps, K-maps for product of sum form, minimize Boolean expression using K-map and obtain K-map from Boolean expression, Quine Mc Cluskey Method. | 12 |

| | | |
|---|---|---|
| **III** | **Combinational Logic Circuits:**<br>Introduction, Multi-input, multi-output Combinational circuits, Code converters design and implementations<br>**Arithmetic Circuits:**<br>Introduction, Adder, BCD Adder, Excess – 3 Adder, Binary Subtractors, BCD Subtractor, Multiplier, Comparator. | **12** |
| **IV** | **Multiplexer, Demultiplexer, ALU, Encoder and Decoder:**<br>Introduction, Multiplexer, Demultiplexer, Decoder, ALU, Encoders.<br>**Sequential Circuits: Flip-Flop:**<br>Introduction, Terminologies used, S-R flip-flop, D flip-fop, JK flip-flop, Race-around condition, Master – slave JK flip-flop, T flip-flop | **12** |
| **V** | **Counters:**<br>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.<br>**Shift Register:**<br>Introduction, parallel and shift registers, serial shifting, serial–in serial–out, serial–in parallel–out , parallel–in parallel–out, Ring counter, Johnson counter, Applications of shift registers, Pseudo-random binary sequence generator, IC7495, Seven Segment displays, analysis of shift counters. | **12** |

**Books and References**

| Sr. No. | Title | Author/s | Publisher | Edition | Year |
|---|---|---|---|---|---|
| **1** | Digital Electronics and Logic Design | N. G. Palan | Technova | 1st | 2000 |
| **2** | Make Electronics | Charles Platt | O’Reilly | 1st | 2010 |
| **3** | Modern Digital Electronics | R. P. Jain | Tata McGraw Hill | 4th | 2009 |
| **4** | Digital Principles and Applications | Malvino and Leach | Tata McGraw Hill | 8th | 2014 |
| **5** | Digital Electronics: Principles, Devices and Applications, | Anil K. Maini | Wiley | 1st | 2007 |

**Internal Evaluation: 40 Marks**

| 20 Marks | 20 Marks |
|---|---|
| Class Test | Assignments |

## FYBSc(IT) - Semester I

# Digital Electronics (SIUSITP12)

**List of Practical (Online Simulators like MMLogic, Logisim, CircuitVerse, CircuitLab can be used)**

| | |
|---|---|
| **1.** | **Study of Logic gates and their ICs and universal gates:** |
| | a. Study of AND, OR, NOT, XOR, XNOR, NAND and NOR gates<br>b. IC 7400, 7402, 7404, 7408, 7432, 7486, 74266<br>c. Implement AND, OR, NOT, XOR, XNOR using NAND gates.<br>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.<br>b. Implement other given expressions using minimum number of gates.<br>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.<br>b. Design and implement Gray – to – Binary code converter.<br>c. Design and implement Binary – to – BCD code converter<br>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.<br>b. Design and implement BCD adder.<br>c. Design and implement XS – 3 adder.<br>d. Design and implement binary subtractor.<br>e. Design and implement BCD subtractor.<br>f. Design and implement XS – 3 subtractor. |
| | |
| **6.** | **Implement Arithmetic circuits.** |
| | a. Design and implement a 2-bit by 2-bit multiplier.<br>b. Design and implement a 2-bit comparator. |
| | |
| **7.** | **Implement Encode and Decoder and Multiplexer and Demultiplexers.** |
| | a. Design and implement 8:3 encoder.<br>b. Design and implement 3:8 decoder.<br>c. Design and implement 4:1 multiplexer. Study of IC 74153, 74157<br>d. Design and implement 1:4 demultiplexer. Study of IC 74139<br>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.<br>b. Study of IC 7474.<br>c. Study of IC 7476.<br>d. Conversion of Flip-flops.<br>e. Design of 3-bit synchronous counter using 7473 and required gates.<br>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.<br>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.<br>b. Study of ID 7495.<br>c. Implementation of digits using seven segment displays. |

# Web Programming (SIUSIT13)

## Course Objective:

To impart knowledge on client and server scripting, database technology, so as to successfully design and implement a web site.

## Course Outcomes:

**Upon completion of this course, student will be able to:**

CO1: Describe the concepts and architecture of the World Wide Web.

CO2: Create a basic website using HTML and Cascading Style Sheets

CO3: Design and implement dynamic web page with validation using JavaScript objects and apply different event handling mechanisms.

CO4: Build dynamic website using server-side PHP programming.

CO5: Design frontend and connect to backend databases.

| Unit | Contents | No. of Lectures |
|---|---|---|
| I | **Internet and the World Wide Web:**<br>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<br>**HTML5:**<br>Introduction, Why HTML5? Formatting text by using tags, using lists and backgrounds, Creating hyperlinks and anchors. Style sheets, CSS formatting text using style sheets, formatting paragraphs using style sheets. | **12** |
| II | **HTML5 Page layout and navigation:**<br>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.<br>**HTML5 Tables, Forms and Media:**<br>Creating tables: creating simple table, specifying the size of the table, specifying the width of the column, merging table cells, using tables for page layout, formatting tables: applying table borders, applying background and foreground fills, changing cell padding, spacing and alignment, creating user forms: creating basic form, using check boxes and option buttons, creating lists, additional input types in HTML5, Incorporating sound and video: audio and video in HTML5, HTML multimedia basics, embedding video clips, incorporating audio on web page. | **12** |

| | | |
|---|---|---|
| **III** | **Java Script:** Introduction, Client-Side JavaScript, Server-Side JavaScript, JavaScript Objects, JavaScript Security<br>**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<br>**Statements:** Break, comment, continue, delete, do...while, export, for, for...in, function, if...else, import, labelled, return, switch, var, while, with<br>**Core JavaScript (Properties and Methods of Each) :** Array, Boolean, Date, Function, Math, Number, Object, String, regExp<br>**Document and its associated objects:** document, Link, Area, Anchor, Image, Applet, Layer<br>**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 | **12** |
| **IV** | **PHP:**<br>Why PHP and MySQL? Server-side scripting, PHP syntax and variables, comments, types, control structures, branching, looping, termination, functions, passing information with PHP, GET, POST, formatting form variables, superglobal arrays, strings and string functions, regular expressions, arrays, number handling, basic PHP errors/problems | **12** |
| **V** | **Advanced PHP and MySQL :**<br>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** |

**Books and References**

| **Sr. No.** | **Title** | **Author/s** | **Publisher** | **Edition** | **Year** |
|---|---|---|---|---|---|
| 1. | Web Design The Complete Reference | Thomas Powell | TataMcGraw Hill | 2nd | 2000 |
| 2. | HTML5 Step by Step | FaitheWempen | 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. | Head First HTML 5 programming | Eric Freeman | O’Reilly | | 2013 |

| | | | | | |
|---|---|---|---|---|---|
| 7. | JavaScript 2.0: The Complete Reference | Thomas Powell and Fritz Schneider | TataMcGraw Hill | 2nd | |
| 8. | Murach’s PHP and MySQL | Joel Murach<br>Ray Harris | SPD | | 2011 |

**Internal Evaluation: 40 Marks**

| **20 Marks** | **20 Marks** |
|---|---|
| Class Test | Developing responsive website using HTML, CSS, JavaScript, PHP and MYSQL. |

# FYBSc(IT) - Semester I

## Web Programming (SIUSITP13)

**List of Practical: (To be done using XAMPP server)**

| | |
|---|---|
| **1.** | **Use of Basic Tags** |
| a. | Design a web page using different text formatting tags. |
| b. | Design a web page with links to different pages and allow navigation between web pages. |
| c. | Design a web page demonstrating all Style sheet types |
| | |
| **2.** | **Image maps, Tables, Forms and Media** |
| a. | Design a web page with Imagemaps. |
| b. | Design a web page demonstrating different semantics |
| c. | Design a web page with different tables. Design a webpages using table so that the content appears well placed. |
| d. | Design a web page with a form that uses all types of controls. |
| e. | Design a web page embedding with multimedia features. |
| | |
| **3.** | **Java Script** |
| a. | Using JavaScript design, a web page that prints factorial/Fibonacci series/any given series. |
| b. | Design a form and validate all the controls placed on the form using Java Script. |
| c. | Write a JavaScript program to display all the prime numbers between 1 and 100. |
| a. | Write a JavaScript program to accept a number from the user and display the sum of its digits. |
| d. | 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). |
| e. | Write a java script program to design simple calculator. |
| | |
| **4.** | **Control and looping statements and Java Script references** |
| a. | Design a web page demonstrating different conditional statements. |
| b. | Design a web page demonstrating different looping statements. |
| c. | Design a web page demonstrating different Core JavaScript references (Array, Boolean, Date, Function, Math, Number, Object, String, regExp). |
| | |
| **5.** | **Basic PHP I** |
| a. | Write a PHP Program to accept a number from the user and print it factorial. |
| b. | Write a PHP program to accept a number from the user and print whether it is prime or not. |
| | |
| **6.** | **Basic PHP II** |
| a. | Write a PHP code to find the greater of 2 numbers. Accept the no. from the user. |
| b. | Write a PHP program to display the following Binary Pyramid:<br>1<br>0 1 |

| | |
|---|---|
| | 1 0 1<br>0 1 0 1 |
| | |
| **7.** | **String Functions and arrays** |
| a. | Write a PHP program to demonstrate different string functions. |
| b. | Write a PHP program to create one dimensional array. |
| | |
| **8.** | **PHP and Database** |
| a. | Write a PHP code to create:<br>• Create a database College<br>• Create a table Department (Dname, Dno, Number_Of_faculty) |
| b. | Write a PHP program to create a database named “College”. Create a table named “Student” with following fields (sno, sname, percentage). Insert 3 records of your choice. Display the names of the students whose percentage is between 35 to 75 in a tabular format. |
| c. | Design a PHP page for authenticating a user. |
| | |
| **9.** | **Email** |
| a. | Write a program to send email with attachment. |
| | |
| **10.** | **Sessions and Cookies** |
| a. | Write a program to demonstrate use of sessions and cookies. |

## FYBSc(IT) - Semester I

# Discrete Mathematics (SIUSIT14)

## Course Objective:

To develop the logical and analytical thinking of the student which will familiarize them with the concepts required for the competitive exams

## Course Outcomes:

**Upon completion of this course, student will be able to:**

CO1: Use concepts of set theory , conditional statements and identify valid & invalid arguments.

CO2: Explain the significance of quantified statements

CO3: Describe sequences, mathematical induction and recursion in Mathematics.

CO4: Classify relations, graphs and trees ,implement functions on general sets

CO5: Solve problems related to counting and probability.

| Unit | Contents | No. of Lectures |
|---|---|---|
| I | **Introduction**: Variables, The Language of Sets, The Language of Relations and Function<br>**Set Theory**: Definitions and the Element Method of Proof, Properties of Sets, Disproofs, Algebraic Proofs, Boolean Algebras, Russell's Paradox and the Halting Problem.<br>**The Logic of Compound Statements**: Logical Form and Logical Equivalence, Conditional Statements, Valid and Invalid Arguments | **12** |
| II | **Quantified Statements**: Predicates and Quantified Statements, Statements with Multiple Quantifiers, Arguments with Quantified Statements<br>**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** |
| III | **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.<br>**Functions**: Functions Defined on General Sets, One-to-One and Onto, Inverse Functions, Composition of Functions, Cardinality with Applications to Computability | **12** |

| | | |
|---|---|---|
| **IV** | **Relations**: Relations on Sets, Reflexivity, Symmetry, and Transitivity, Equivalence Relations, Partial Order Relations<br>**Graphs and Trees**: Definitions and Basic Properties, Trails, Paths, and Circuits, Matrix Representations of Graphs, Isomorphism's of Graphs, Trees, Rooted Trees, Isomorphism's of Graphs, Spanning trees and shortest paths. | **12** |
| **V** | **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, rCombinations with Repetition Allowed, Probability Axioms and<br>Expected Value, Conditional Probability, Bayes' Formula, and Independent Events. | **12** |

**Books and References**

| **Sr. No.** | **Title** | **Author/s** | **Publisher** | **Edition** | **Year** |
|---|---|---|---|---|---|
| 1. | Discrete Mathematics with Applications | Sussana S. Epp | Cengage Learning | 4th | 2010 |
| 2. | Discrete Mathematics, Schaum's Outlines Series | Seymour Lipschutz, Marc Lipson | Tata MCGraw Hill | 3rd | 2017 |
| 3. | Discrete Mathematics and its Applications | Kenneth H. Rosen | Tata MCGraw Hill | 7th | 2017 |
| 4. | Discrete mathematical structures | B Kolman RC Busby, S Ross | PHI | 6th | 2015 |
| 5. | Discrete structures | Liu | Tata MCGraw Hill | 3rd | 2008 |

**Internal Evaluation: 40 Marks**

| **20 Marks** | **20 Marks** |
|---|---|
| Class Test | Assignment(problem solving) |

# FYBSc(IT) - Semester I

## Discrete Mathematics Practical (SIUSITP14)

**List of Practical: (To be done in 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 |

# FYBSc(IT) - Semester I

## Communication Skills (SIUSIT15)

### Course Objective:

To improve the communication skills of learners and to introduce them to the various modes of business communication.

### Course Outcomes:

**Upon completion of this course, student will be able to:**

CO1: Describe the concept of communication along with the Seven C's of effective communication.

CO2: Write letters, E-mails, memos, notice, agenda, brochures.

CO3: Write business reports, abstracts and summaries

CO4: Develop reading , listening and oral communication skills

CO5: Describe the mechanics of writing like Transitions, Spelling Rules, Hyphenation etc.

| Unit | Contents | No. of Lectures |
|---|---|---|
| I | **The Seven Cs of Effective Communication:**<br>Completeness, Conciseness, Consideration, Concreteness, Clarity, Courtesy, Correctness<br>Basics of Communication, Non-Verbal Communication, Barriers to Communication | 12 |
| II | **Business Communication at Workplace:**<br>Letter Components and Layout, Planning a Letter, Process of Letter Writing, E-Mail Communication, Memo and Memo Reports, Employment Communication, Notice, Agenda and Minutes of Meeting, Brochures. | 12 |
| III | **Report Writing:**<br>Effective Writing, Types of Business Reports, Structure of Reports, Gathering Information, Organisation of Material, Writing Abstracts and Summaries, Visual Aids | 12 |
| IV | **Required Skills:**<br>Reading Skill, Listening Skill, Note-Making, Precis Writing, Audio-Visual Aids, Oral Communication. | 12 |
| V | **Mechanics of Writing:**<br>Transitions, Spelling Rules, Hyphenation, Transcribing Numbers, Proof Reading | 12 |

**Books and References**

| Sr. No. | Title | Author/s | Publisher | Edition | Year |
|---|---|---|---|---|---|
| **1** | Business Communication | Meenakshi Raman and Prakash Singh | Oxford University Press | 2nd | 2012 |
| **2** | Professional Communication | ArunaKoneru | Tata McGraw Hill | | 2008 |
| **3** | Effective Business Communication | Herta A Murphy, Herbert Hildebrant, Jane P Thomas | Tata McGraw Hill | 7th | 2017 |

**Internal Evaluation: 40 Marks**

| 20 Marks | 20 Marks |
|---|---|
| Class Test | Book review ( A book on any appropriate subject ) |

# FYBSc(IT) - Semester I

## Communication Skills Practical (SIUSITP15)

**List of Practical:**

| | |
|---|---|
| **1.** | **Communication Origami** |
| | Give one sheet of letter size/A4 paper to each<br>Tell the group that you will start giving them all instructions on how to fold the paper to create an origami shape.<br>Tell the group that while you give them the instructions, they must keep their eyes closed and cannot ask any questions<br>Start giving the group several instructions to fold and rip their paper several times then ask them to unfold their paper and compare how it looks like.<br>**Debrief and discussion:**<br>Make the point that each paper looks different even though you have given the same instructions to everybody. What does this mean?<br>Ask the group if you think the results would have been better if they kept their eyes open or were allowed to ask questions<br>Communicating clearly is not easy, we all interpret the information we get differently that's why it's very important to ask questions and confirm understanding to ensure the communicated message is not distorted. |
| **2.** | **Guessing Game** |
| | Split your class into two equal groups/teams<br>One person from each team think of a business object (any common business object that can be found in any office like a stapler, printer, ..etc.)<br>When each person returns to his team, it's the team's task to ask him/her closed ended questions only to try and find out what the object is. If needed, explain that closed ended questions are those that can be answered by yes or no<br>Once any team finds the object, this means that they won this round. And they can go for another round.<br>**Discussion and debrief:**<br>Tell the group that obviously it took a long time and effort for us to find out the object in each round, but what if we had not time and only had one question to ask to find out the object, what would that question be?<br>The question would be "What is the object?" which is an open ended question.<br>Open ended questions are an excellent way to save time and energy and helps you get to the information you need fast, however closed questions can also be very useful in some instances to confirm your understanding or to help you control the conversation with an overly talkative person/customer. |
| **3.** | **Guessing the emotion** |
| | Participants are split into teams and act out an emotion, such as disgust, affection, fear, anxiety, embarrassment, anger, determination, etc. and the total group will try to guess |

| | |
|---|---|
| | what the emotion is.<br>Divide the group into two teams.<br>Place on a table (or put in a box) a packet of cards, each of which has a particular emotion typed on it.<br>Have a participant from Group A take the top card from the table and act out (pantomime) the emotion for his/her group. This is to be done in a fixed time limit (such as a minute or two).<br>If the emotion is guessed correctly by Group A, they receive ten points.<br>Now have a participant from Group B act out an emotion; award points as appropriate.<br>Rotate the acting opportunities between the two groups.<br>After 20–30 minutes, call time and announce the winning team based on its point total.<br>**Discussion and debrief:**<br>Facial expressions Gestures contribute greatly to communication. You do not need words to convey message. |
| **4.** | **Body Language** |
| | Body language speaks louder than any words you can ever utter. Whether you're telling people that you love them, you're angry with them, or don't care less about them, your body movements reveal your thoughts, moods, and attitudes. Both consciously and sub-consciously your body tells observers what's really going on with you.<br>Explain to the group that you are going to give them a series of instructions, which you would like them to copy as fast as they can<br>State the following actions as YOU do them:<br>• Put your hand to your nose<br>• Clap your hands<br>• Stand up<br>• Touch your shoulder<br>• Sit down<br>• Stamp your foot<br>• Cross your arms<br>• Put your hand to your mouth – BUT WHILE SAYING THIS PUT YOUR HAND TO YOUR NOSE<br>**Discussion and debrief:**<br>Observe the number of group members who copy what you did rather than what you said. |
| **5.** | **Brainstorming session** |
| | The participants will be divided into groups of 3-4 . A day before the practical session, the participants will be instructed to bring stationary: colored card board papers, book mark colored slips , fevistick, colored sketch pens .<br>A topic will be given to the students. They will be given a topic and they will be told to write down whatever words that come to their mind , on the bookmarks lips and stick on the card board. Group members should not discuss about any word or its relevance to the topic. Whatever comes to their mind, they should just note down and stick. Also members should not restrict other members on any concept/word. This activity is to be completed in limited time frame (eg. 10 mins)<br>**Discussion and debrief:**<br>Different members in the same group will come up with different ideas even though the topic was same. So each person think / communicates differently. Everyone's thinking |

| | |
|---|---|
| | procedure differs. Also communication/ thinking procedure depends on linguistic factors like vocabulary, understanding of the topic and fluency in language . |
| **6.** | **On the spot Elocution / Debate** |
| | The class will be divided into two groups and a topic will be given on the spot for debating.<br>**Discussion and debrief:**<br>**The stage fear gets conquered. The fluency in language matters. Students will come to know that reading books helps. The skill of putting their point firmly and convince others will get tested.** |
| | |
| **7.** | **Group Discussion** |
| | The class will be divided into two groups and a topic will be given on the spot for debating. The participants are expected to abide by the rules and protocol of the group discussion.<br>**Discussion and debrief:**<br>**To actively participate student must be aware of day to day happening. Their general knowledge will get tested. They will be convinced that reading daily newspaper or watching news, keeping track of all important events in social, political aspects is necessary. They will learn to put their points. Their convincing skills, logical thinking, team spirit would enhance.** |
| **8.** | **Use of word processing software like Microsoft Office /Libre Office for communication** |
| | A letter / report is to be made. The different features of the software are exploited. |
| **9.** | **Use of word processing software like Microsoft Office /Libre Office for communication** |
| | A spreadsheet is made. The different features of the software are exploited. |
| **10.** | **Presentation software like Microsoft Power Point for Communication.** |
| | PPT presentations to be done. Images, Sound to be included in PPTs. |
| **11.** | **Memory Test.** |
| | This can be a great activity for presentation skills or train the trainer classes. Tell participants that you will read them a list of words to test their memory. Participants will need to listen carefully and cannot write any of the words you will say down. Later you will test and see how many words they still remember.<br>Read each of the following words slowly and pause briefly between each word. Note that one of the words (nigh) is repeated three times.<br>**dream**, **sleep**, **night**, **mattress**, **snooze**, **sheet**, **nod**, **tired**, **night**, **artichoke**, **insomnia**, **blanket**, **night**, **alarm**, **nap**, **snore**, **pillow**, **alarm**<br><br>Once you finish reading the list, try to distract them by talking about anything else for about one minute. Then ask each participant to take out a piece of paper and write down as many words as they can remember.<br>**Debrief by exploring the four basic principles of memory as follows:**<br>**Primacy and recency** – ask participants to raise their hands if they remembered the first and last words (dream and pillow). Explain that people easily remember the first and last things they hear in a series. Link back to the importance of having a high energy start and a final recap and review of your presentation. |

| | |
|---|---|
| | **Surprise** – ask those who remember the word (artichoke) to raise their hands. Make the point that most people tend to remember things that are different, new or unexpected. People will remember your presentation for much longer if it is novel and untraditional if not shocking.<br>**Repetition** – Ask those who remember the word (night) to raise their hands. Most participants must have remembered and wrote this word because you repeated it three times. Explain that people remember things more if they are repeated and how important it is to recap and review the main key points of your presentation more than once to ensure your audience can remember them.<br>**False-memory** – Ask participants to raise their hands if they remember the word (bed). Reveal that this word was not in the list but still some of them did write it down and raise their hands. Explain that our brain automatically closes gaps in what it sees and hears or reads, and sometimes assumes things that never took place happened. Most participants would have written the word (bed) because it simply fits and belongs to the list logically even though you never read it. |

# FYBSc(IT) - Semester II

| Course Code | Course Type | Course Title | Periods per week (1 Period = 50 min) | Credits |
|---|---|---|---|---|
| SIUSIT21 | Core Subject | Object oriented Programming | **5** | **2** |
| SIUSIT22 | Core Subject | Microprocessor Architecture | **5** | **2** |
| SIUSIT23 | Core Subject | Operating Systems | **5** | **2** |
| SIUSIT24 | Core Subject | Numerical and Statistical Methods | **5** | **2** |
| SIUSIT25 | Ability Enhancement Skill Course | Green Computing | **5** | **2** |
| | | | | |
| SIUSITP21 | Core Subject Practical | Object Oriented Programming Practical | **3** | **2** |
| SIUSITP22 | Core Subject Practical | Microprocessor Architecture Practical | **3** | **2** |
| SIUSITP23 | Core Subject Practical | Operating Systems Practical | **3** | **2** |
| SIUSITP24 | Core Subject Practical | Numerical and Statistical Methods Practical | **3** | **2** |
| SIUSITP25 | Ability Enhancement Skill Course Practical | Green Computing Practical | **3** | **2** |
| **TOTAL CREDITS** | | | | **20** |

<u>**FYBSc(IT) - Semester II**</u>

# Object Oriented Programming (SIUSIT21)

## Course Objective:

To orient the learners about the principles behind object-oriented programming, its benefits and application in real world scenarios.

## Course Outcomes:

**Upon completion of this course, student will be able to:**

CO1: Explain the difference between top down and bottom up approach in programming and outline the essential features and elements of C++ programming

CO2: Apply the concepts of member functions, constructors, destructors in C++.

CO3: Describe and apply the concepts of function and operator overloading and inheritance.

CO4: Incorporate exception handling in object oriented programs.

CO5: Use template classes and standard library in C++

| Unit | Contents | No. of Lectures |
|---|---|---|
| I | **Introduction:** Object oriented and Procedure oriented Language, Object oriented theme, development, benefits and applications of OOPs<br>**Principles of OOPS:** OOPS Paradigm, Basic Concepts of OOPS: Objects, Classes, Data Abstraction and Data Encapsulation, Inheritance, Polymorphism, Dynamic Binding, Message Passing | 12 |
| II | **Classes and Objects:** Simple Class structure, Defining member functions inside and outside class, passing object as an argument, Returning object from functions, friend function, Pointer to object, Array of pointer to object<br>**Constructors and Destructors:** Introduction, Default Constructor, Parameterized Constructor and examples, copy constructor, Destructors | 12 |
| III | **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<br>**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. | 12 |
| IV | **Virtual Functions & Abstract Class:** Introduction and need, Pure Virtual Functions, Static Functions, this Pointer, abstract classes, virtual destructors<br>**String Handling:** Introduction, creating string objects, string characteristics, manipulating string | 12 |

|  |  |  |
|---|---|---|
|  | **Exception Handling:** Introduction, Exception Handling Mechanism, Concept of throw & catch with example |  |
| **V** | **Templates:** Introduction, Function Template and examples, Class Template and examples.<br>**Working with Files:** Introduction, File Operations, Various File Modes, File Pointer and their Manipulation | **12** |

**Books and References**

| **Sr. No.** | **Title** | **Author/s** | **Publisher** | **Edition** | **Year** |
|---|---|---|---|---|---|
| **1** | Object Oriented Analysis and Design | Timothy Budd | TMH | 3rd | 2012 |
| **2** | Mastering C++ | K R Venugopal, RajkumarBuyya, T Ravishankar | TMH | 2nd | 2011 |
| **3** | Object Oriented Programming with C++ | E Balagurusamy | TMH | 7th | 2017 |
| **4** | C++ for beginners | B. M. Hirwani | SPD | -- | 2013 |
| **5** | Effective Modern C++ | Scott Meyers | SPD | -- | 2014 |
| **6** | C++ Primer | Stanley. B, Josee Lajoi, Barbara Moo | Addison Wesley | 5th | 2012 |

**Internal Evaluation: 40 Marks**

| **20 Marks** | **20 Marks** |
|---|---|
| Class Test | Projects to be developed in C++ |

# FYBSc(IT) - Semester II

## Object Oriented Programming Practical (SIUSITP21)

**List of Practical: (To be done in C++)**

| | |
|---|---|
| 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<br>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.WheregetData() will be private method.<br>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.WherereadNo() will be private method.<br>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<br>b. Write a friend function for adding the two different distances and display its sum, using two classes.<br>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<br>b. Design a class Geometry containing the methods area() and volume() and also overload the area() function.<br>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<br>b. Overload the operator + for adding the timings of two clocks, And also pass objects as an argument.<br>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.<br>b. Design a class for multiple inheritance.<br>c. Implement the hierarchical inheritance |

| | |
|---|---|
| 6. | **Virtual functions and abstract classes** |
| | a. Implement the concept of method overriding.<br>b. Show the use of virtual function<br>c. Show the implementation of abstract class. |
| 7. | **String handling** |
| | a. String operations for string length , string concatenation<br>b. String operations for string reverse, string comparison,<br>c. Console formatting functions. |
| 8. | **Exception handling** |
| | a. Show the implementation of exception handling<br>b. Show the implementation for exception handling for strings<br>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.<br>b. Design a class to handle multiple files and file operations<br>c. Design a editor for appending and editing the files |
| 10. | **Templates** |
| | a. Show the implementation for template function<br>b. Show the implementation of template class library for swap function.<br>c. Design the template class library for sorting ascending to descending and vice versa |

## FYBSc(IT) - Semester II

# Microprocessor Architecture (SIUSIT22)

## Course Objective:

To orient and take learners through a captivating journey of basic programming language of 8085 microprocessor so that it helps them to have an insight with the internal working and operations performed by the microprocessor.

## Course Outcomes:

**Upon completion of this course, student will be able to:**

CO1: Compare and contrast microprocessor and microcomputers and analyze the Assembly language, Memory interface and Microprocessor Architecture.

CO2: Analyze the interfacing of I/O devices , inspect writing and debugging of 8085 assembly language programs by using various addressing modes.

CO3: Demonstrate and apply various programming techniques to create counter and time delay .

CO4: Analyze, demonstrate and apply the stack operations and code conversions .

CO5: Analyze assembly language programs and select appropriate assembler to run 8085 programs on it with the help of a cross assembler

| Unit | Contents | No. of Lectures |
|---|---|---|
| I | **Microprocessor, microcomputers, and Assembly Language:**<br>Microprocessor, Microprocessor Instruction Set and Computer Languages, From Large Computers to Single-Chip Microcontrollers, Applications.<br>**Microprocessor Architecture and Microcomputer System:**<br>Microprocessor Architecture and its operation's, Memory, I/O Devices, Microcomputer System, Logic Devices and Interfacing, Microprocessor-Based System Application.<br>**8085 Microprocessor Architecture and Memory Interface:**<br>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. | 12 |
| II | **Interfacing of I/O Devices**<br>Basic Interfacing concepts, Interfacing Output Displays, Interfacing Input Devices, Memory Mapped I/O, Testing and Troubleshooting I/O Interfacing Circuits.<br>**Introduction to 8085 Assembly Language Programming:**<br>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.<br>**Introduction to 8085 Instructions:** | 12 |

| | | |
|---|---|---|
| | Data Transfer Operations, Arithmetic Operations, Logic Operation, Branch Operation, Writing Assembly Languages Programs, Debugging a Program. | |
| **III** | **Programming Techniques With Additional Instructions:**<br>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.<br>**Counters and Time Delays:**<br>Counters and Time Delays, Illustrative Program: Hexadecimal Counter, Illustrative Program: zero-to-nine (Modulo Ten) Counter, Generating Pulse Waveforms, Debugging Counter and Time-Delay Programs. | **12** |
| **IV** | **Stacks and Sub-Routines:**<br>Stack, Subroutine, Restart, Conditional Call, Return Instructions, Advanced Subroutine concepts.<br>**Code Conversion, BCD Arithmetic, and 16-Bit Data Operations:**<br>BCD-to-Binary Conversion, Binary-to-BCD Conversion, BCD-to Seven-Segment-LED Code Conversion, Binary-to-ASCII and ASCII to-Binary Code Conversion, BCD Addition, BCD Subtraction, Introduction To Advanced Instructions and Applications, Multiplication, Subtraction With Carry. | **12** |
| **V** | **Software Development System and Assemblers:**<br>Microprocessors-Based Software Development system, Operating System and Programming Tools, Assemblers and Cross-Assemblers, Writing Program Using Cross Assemblers.<br>**Interrupts:**<br>The 8085 Interrupt, 8085 Vectored Interrupts, Restart as S/W Instructions, Additional I/O Concepts and processes.<br>**Introduction to recent microprocessor:** Core2,i3,SUN SPARC Microprocessor, Rasberry Pi 4, Arduino | **12** |

**Books and References**

| **Sr. No.** | **Title** | **Author/s** | **Publisher** | **Edition** | **Year** |
|---|---|---|---|---|---|
| 1. | Microprocessors Architecture, Programming and Applications with the 8085. | Ramesh Gaonkar | PENRAM | 6th | 2016 |
| 2. | Computer System Architecture | M. Morris Mano | PHI | 3rd | 2007 |
| 3. | Structured Computer Organization | Andrew C. Tanenbaum | PHI | 6th | 2016 |
| 4. | 8080A/8085 Assembly Language Programming | Lance A. Leventhel | McGraw Hill | 1st | 1978 |

**Internal Evaluation: 40 Marks**

| **20 Marks** | **20 Marks** |
|---|---|
| Class Test | Assignments |

## FYBSc(IT) - Semester II

# Microprocessor Architecture Practical (SIUSITP22)

| 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 numbers stored in memory locations 4200H and 4201H and store result in memory location 4300H. Assume the least significant digit is stored at 4200H. |
| b. | Two digit BCD number is stored in memory location 4200H. Unpack the BCD number and store the two digits in memory locations 4300H and 4301H such that memory location 4300H will have lower BCD digit. |
| | |
| **4.** | **Register Operations.** |
| a. | Write a program to shift an eight bit data four bits right. Assume that data is in register C. |
| b. | Program to shift a 16-bit data 1 bit left. Assume data is in the HL register pair |
| c. | Write a set of instructions to alter the contents of flag register in 8085. |
| d. | Write a program to count number of l's in the contents of D register and store the count in the B register. |

| | |
|---|---|
| **5.** | **Multiple memory locations.** |
| a. | Calculate the sum of series of numbers. The length of the series is in memory location 4200H and the series begins from memory location 4201H. a. Consider the sum to be 8 bit number. So, ignore carries. Store the sum at memory location 4300H.<br>b. Consider the sum to be 16 bit number. Store the sum at memory locations 4300H and 4301H |
| b. | Multiply two 8-bit numbers stored in memory locations 2200H and 2201H by repetitive addition and store the result in memory locations 2300H and 2301H. |
| c. | 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 of the list is 2000H.<br>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:<br>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. | Multiply the 8-bit unsigned number in memory location 2200H by the 8-bit unsigned number in memory location 2201H. Store the 8 least significant bits of the result in memory location 2300H and the 8 most significant bits in memory location 2301H. |
| d. | 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 |

## FYBSc(IT) - Semester II

# Operating Systems (SIUSIT23)

## Course Objective:

To orient the students about the role of operating systems in resource (processor, memory, file and disk) management.

## Course Outcomes:

**Upon completion of this course, student will be able to:**

CO1: Analyze various scheduling algorithms.

CO2: Compare and contrast various memory management schemes.

CO3: Explain the deadlock, prevention and avoidance algorithms.

CO4: Describe the virtual machines and basics of virtualization techniques.

CO5: Compare Linux, Android and Windows operating systems.

| Unit | Contents | No. of Lectures |
|---|---|---|
| I | **Introduction:**<br>What is an operating system? History of operating system, computer hardware, different operating systems, operating system concepts, system calls, operating system structure.<br>**Processes and Threads:**<br>Processes, threads, interprocess communication, scheduling, IPC problems. | 12 |
| II | **Memory Management:**<br>No memory abstraction, memory abstraction: address spaces, virtual memory, page replacement algorithms, design issues for paging systems, implementation issues, segmentation.<br>**File Systems:**<br>Files, directories, file system implementation,file-system management and optimization, MS-DOS file system, UNIX V7 file system, CD ROM file system. | 12 |
| III | **Input-Output:**<br>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.<br>**Deadlocks:**<br>Resources, introduction to deadlocks, the ostrich algorithm, deadlock detection and recovery, deadlock avoidance, deadlock prevention, issues. | 12 |

| | | |
|---|---|---|
| **IV** | **Virtualization and Cloud:**<br>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.<br>**Multiple Processor Systems**<br>Multiprocessors, multicomputers, distributed systems. | **12** |
| **V** | **Case Study on LINUX and ANDROID:**<br>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<br>**Case Study on Windows:**<br>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. | **12** |

**Books and References**

| **Sr. No.** | **Title** | **Author/s** | **Publisher** | **Edition** | **Year** |
|---|---|---|---|---|---|
| 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 System Concepts | Abraham Silberschatz, Peter B. Galvineg Gagne | Wiley | 9th | 2012 |
| 4. | Operating Systems | Godbole and Kahate | McGraw Hill | 3rd | 2017 |

**Internal Evaluation: 40 Marks**

| **20 Marks** | **20 Marks** |
|---|---|
| Class Test | Presentation on various topics of Operating Systems |

## FYBSc(IT) - Semester II

# Operating Systems Practical (SIUSITP23)

**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** |

# FYBSc(IT) - Semester II

## Numerical and Statistical Methods (SIUSIT24)

### Course Objective:

The learner will be exposed to the development, computation and application of optimal control algorithms required for scientific computing and data analysis.

### Course Outcomes:

**Upon completion of this course, student will be able to:**

CO1: Identify the role of errors , solve algebraic and transcendental equations

CO2: Apply various interpolation to find an unknown value from the set of given values related to a situation

CO3: Find the solution of simultaneous algebraic equations using iterative methods , apply differentiation and integration using various rules like Trapezoidal Rule, Simpson's Rule, Euler's Method, Runge-Kutta Method.

CO4: Apply Linear Regression and Linear Programming Problems for any real life situation

CO5: Compare the role of various distributions such as Uniform, Binomial, Poisson and Bernoulli.

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

| | | |
|---|---|---|
| **IV** | **Least-Squares Regression:**<br>Linear Regression, Polynomial Regression, Multiple Linear Regression, General Linear Least Squares, Nonlinear Regression<br>**Linear Programming:**Linear optimization problem, Formulation and Graphical solution, Basic solution and Feasible solution. | **12** |
| **V** | **Random variables:** Discrete and Continuous random variables, Probability density function, Probability distribution of random variables, Expected value, Variance.<br>**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** |

**Books and References**

| **Sr. No.** | **Title** | **Author/s** | **Publisher** | **Edition** | **Year** |
|---|---|---|---|---|---|
| **1** | Introductory Methods of Numerical Methods | S. S. Shastri | PHI | 5th | 2012 |
| **2** | Numerical Methods for Engineers | Steven C. Chapra, Raymond P. Canale | Tata Mc Graw Hill | 6th | 2010 |
| **3** | Numerical Analysis | Richard L. Burden, J. Douglas Faires | Cengage Learning | 9th | 2011 |
| **4** | Fundamentals of Mathematical Statistics | S. C. Gupta, V. K. Kapoor | Sultan Chand & Sons | 12th | 2020 |
| **5** | Elements of Applied Mathematics | P.N.Wartikar and J.N.Wartikar | A. V. Griha, Pune | Volume 1 and 2 | |

**Internal Evaluation: 40 Marks**

| **20 Marks** | **20 Marks** |
|---|---|
| Class Test | Assignments (problem solving) |

# FYBSc(IT) - Semester II

## Numerical and Statistical Methods Practical (SIUSITP24)

**List of Practical:**

| | |
|---|---|
| **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 $e^x$ 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. | Programming to obtain derivatives numerically. |
| | |
| **6.** | **Numerical Integration** |
| a. | Program for numerical integration using Trapezoidal rule. |
| b. | Program for numerical integration using Simpson's 1/3rdrule. |
| c. | Program for numerical integration using Simpson's 3/8thrule. |
| | |
| **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 (SIUSIT25)

## Course Objective:

To make learners aware of the green computing practices that can be used to minimize negative impacts on the environment.

## Course Outcomes:

**Upon completion of this course, student will be able to:**

CO1: Explain the importance of Green IT & some issues related to it.

CO2: Illustrate the use of cooling and minimizing power usage.

CO3: Find how to recycle e-waste, reduce paper waste and carbon footprint.

CO4: Describe the importance of the use of environmentally sustainable computers and electronic systems.

CO5: Examine the various global standards and initiatives in green computing.

| Unit | Contents | No. of Lectures |
|---|---|---|
| I | **Green IT Overview:** Introduction , Environmental Concerns and Sustainable Development, Environmental Impacts of IT, Green I , Holistic Approach to Greening IT, Greening IT, Applying IT for Enhancing Environmental Sustainability, Green IT Standards and Eco-Labelling of IT , Enterprise Green IT Strategy, Green Washing, Green IT: Burden or Opportunity?<br>**Problems:** Toxins, Power Consumption, Equipment Disposal, Company's Carbon Footprint: Measuring, Details, reasons to bother, Plan for the Future, Cost Savings: Hardware, Power. | 12 |
| II | **Minimizing Power Usage:**<br>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.<br>**Cooling:**<br>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 |

| | | |
|---|---|---|
| **III** | **Changing the Way of Work:**<br>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.<br>**Going Paperless:**<br>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 | **12** |
| **IV** | **Green Devices and Hardware:** Introduction , Life Cycle of a Device or Hardware, Reuse, Recycle and Dispose<br>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,<br>Using Remote Desktop, Establishing a Connection | **12** |
| **V** | **Greening Your Information Systems:**<br>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.<br>**Staying Green:**<br>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** |

**Books and References**

| Sr. No. | Title | Author/s | Publisher | Edition | Year |
|---|---|---|---|---|---|
| **1** | Green IT | Toby Velte, Anthony Velte, Robert Elsenpeter | McGraw Hill | | 2008 |
| **2** | Harnessing Green IT: Principles and Practices | San Murugesan, G. R. Ganadharan | Wiley & IEEE. | 1st | 2014 |
| **3** | Green Data Center: Steps for the Journey | Alvin Galea, Michael Schaefer, Mike Ebbers | Shroff Publishers and Distributers | | 2018 |
| **4** | Green Computing and Green IT Best Practice | Jason Harris | Emereo | | 2008 |

|  |  |  |  |  |  |
|---|---|---|---|---|---|
| **4** | Green Computing Tools and Techniques for Saving Energy, Money and Resources | Bud E. Smith | CRC Press |  | 2014 |

**Internal Evaluation: 40 Marks**

| **20 Marks** | **20 Marks** |
|---|---|
| Class Test | Case Studies/Presentations |

# FYBSc(IT) - Semester II

## Green Computing Practical (SIUSITP25)

**List of Practical:**

| | **Project and Viva Voce** |
|---|---|
| a. | A project should be done based on the objectives and concepts of Green Computing. |
| b. | The project can be done individually or a group of two students. |
| c. | The students will have to present the project during the examination.(a working model can be included) |
| d. | A certified copy of the project report is essential to appear for the examination. |
| e. | 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. |