

# **SEMESTER II**

## Annexure I

|  |                           |                             |              |
|--|---------------------------|-----------------------------|--------------|
| <b>B. Sc (Information Technology)</b>                    |                           | <b>Semester – I</b>         |              |
| <b>Course Name: Object Oriented Programming with C++</b> |                           | <b>Course Code: USIT201</b> |              |
| <b>Periods per week (1 Period is 50 minutes)</b>         |                           | <b>5</b>                    |              |
| <b>Credits</b>   |                           | <b>2</b>                    |              |
|  |                           | <b>Hours</b>                | <b>Marks</b> |
| <b>Evaluation System</b>                                 | <b>Theory Examination</b> | <b>2</b>                    | <b>75</b>    |
|  | <b>Internal</b>           | <b>--</b>                   | <b>25</b>    |

### Course Objectives:

Understand object oriented programming and advanced C++ concepts

- **Be able to explain the difference between object oriented programming and procedural programming.**
- **Be able to program using more advanced C++ features such as composition of objects, operator overloads, dynamic memory allocation, inheritance and polymorphism, file I/O, exception handling, etc.**
- **Be able to build C++ classes using appropriate encapsulation and design principles**
- **Be able to apply object oriented or non-object oriented techniques to solve**
- **bigger computing problems**

| Unit      | Details   | Lectures  |
|-----------|---|-----------|
| <b>I</b>  | <p><b>INTRODUCTION OF OBJECT-ORIENTED DESIGN</b> : Introduction, Objects, Class and Instance, Polymorphism, Inheritance, Object-Oriented Analysis , Finding the Objects ,Conceptual Modeling Requirements Model , Analysis Model, The Design Model, The Implementation Model , Test Model, Object-Oriented Analysis and Design, The Evolution of Object Model, Object-Oriented Programming, Object-Oriented Design, Object-Oriented Analysis, Elements of Object Model ,The Role of OOAD in the Software Life Cycle, OOAD Methodologies, Grady Booch Approach,</p> <p><b>STARTING WITH C++:</b> C++ Overview, C++ Character Set, C++ Tokens, Variables, Counting Tokens, Data Types, Qualifiers, Range of Data Types, Your First C++ Program, Structure of a C++ Program, Styles of, Writing C++ Programs, Programming Examples</p> <p><b>FEATURES OF C++:</b> Introduction, Operators and Expressions, Declaring Constants, Type Conversion, Decision Making: An Introduction, Unconditional Branching Using Goto , Introduction to Looping</p> <p><b>OPERATORS AND REFERENCES IN C++:</b> Introduction, Scope Resolution Operator, Reference Variables, The Bool Data Type, The Operator New and Delete, Malloc Vs. New ,Pointer Member Operators</p> | <b>12</b> |
| <b>II</b> | <p><b>FUNCTION IN C++:</b> Introduction ,Function Declaration/Prototyping ,The Main Function in C++,Recursion ,Call by Reference ,Call by Reference Vs Call by Address , Return by Reference ,,Inline Function ,Function Overloading ,Function with Default Arguments</p> <p><b>CLASS AND OBJECTS IN C++</b> : Working with Class, Structure in C++ ,Accessing Private Data Passing and Returning Object ,Array of Object ,Friend Function ,Static Class Members ,Constant Member Function</p> <p><b>WORKING WITH CONSTRUCTOR AND DESTRUCTOR:</b> Introduction, Constructor with Parameters, Implicit and Explicit Call to Constructor, Copy Constructor, Dynamic Initialization of Objects, Dynamic Constructor, Destructor</p> <p><b>WORKING WITH OPERATOR OVERLOADING:</b> Introduction, Operator Overloading with Binary Operator Overloading Assignment (=) Operator, Overloading</p>  | <b>12</b> |

## Annexure I

|            |  |           |
|------------|--|-----------|
|            | Unary Operators, Overloading Using Friend Function, Rules of Operator Overloading, Type Conversion   |           |
| <b>III</b> | <p><b>WORKING WITH INHERITANCE IN C++:</b> Introduction, Types of Inheritance, Public, Private and Protected Inheritance, Multiple Inheritance, Hierarchical Inheritance, Virtual Base Class, Constructor and Destructor in Inheritance, Containership</p> <p><b>POINTERS TO OBJECTS AND VIRTUAL FUNCTIONS:</b> Pointer to Objects, The This Pointer, What is Binding in C++? , Virtual Functions ,Working of a Virtual Function ,Rules for Virtual Function ,Pure Virtual Function and Abstract Class ,Object Slicing ,Some Facts about Virtual Function ,Virtual Destructor</p> <p><b>INPUT-OUTPUT AND MANIPULATORS IN C++:</b> Introduction, C++ Stream Classes, Unformatted Input/Output, Formatted Input /Output Operations, Manipulators</p> | <b>12</b> |
| <b>IV</b>  | <p><b>FILE HANDLING IN C++:</b> Introduction, File Streams, Opening and Closing a File, File Opening Modes Checking End of File, Random Access in File, Command Line Arguments, Working with Binary Mode Error Handling</p> <p><b>TEMPLATE PROGRAMMING:</b> Introduction , Function Template , Class Template</p> <p><b>EXCEPTION HANDLING IN C++ :</b> Introduction , Basics of Exception Handling , Exception Handling Mechanism , Programming Examples ,Exception Handling with Class Catching all Exceptions , Specifying Exception for a Function</p>   | <b>12</b> |
| <b>V</b>   | <p><b>INTRODUCTION TO THE STANDARD TEMPLATE LIBRARY :</b> Introduction , Components of STL , Containers , Algorithms , Iterators ,Application of Container Classes Function Objects</p> <p><b>MANIPULATING STRINGS :</b>Introduction , Creating (string) Objects , Manipulating String Objects , Relational Operations, String Characteristics , Accessing Characters in Strings , Comparing and Swapping</p> <p><b>NEW FEATURES OF ANSI C++ STANDARD :</b> Introduction ,New Data Types , New Operators , Class Implementation , Namespace Scope , Operator Keywords , New Keywords , New Headers</p>   | <b>12</b> |

| Sr. No. | Title  | Author/s                                 | Publisher                | Edition                  | Year |
|---------|--|--|--------------------------|--------------------------|------|
| 1.      | Object-oriented Programming C++ Simplified       | Hari Mohan Pandey                        | University Science Press | 1 <sup>st</sup> Edition  | 2017 |
| 2.      | Object Oriented Programming in C++               | E Balagurusamy                           | Tata McGraw-Hill         | 5 <sup>th</sup> Edition  | 2011 |
| 3.      | Object-Oriented Programming in C++               | Robert Lafore                            | Sams                     | 4 <sup>th</sup> Edition  | 2002 |
| 4.      | Programming with ANSI C++                        | Bhushan Trivedi                          | Oxford University Press  | 2 <sup>nd</sup> Edition  | 2012 |
| 5.      | Demystified Object-Oriented Programming with C++ | Dorothy R. Kirk                          | Packt Publishing Lt      | 1 <sup>st</sup> Edition  | 2021 |
| 6.      | C++ Programming: An Object-Oriented Approach     | Behrouz A. Forouzan , Richard F. Gilberg | McGraw-Hill Education    | 1 <sup>st</sup> edition  | 2020 |
| 7.      | C++ How to Program                               | Paul Deitel, Harvey Deitel               | Pearson Education        | 10 <sup>th</sup> Edition | 2017 |

# Annexure I

## Course Outcomes:

Learners will be able to,

1. Understand the concept of OOPs, feature of C++ language.
2. Understand and apply various types of Datatypes, Operators, Conversions while designing the program.
3. Understand and apply the concepts of Classes & Objects, friend function, constructors & destructors in program design.
4. Design & implement various forms of inheritance, String class, calling base class constructors.
5. Apply & Analyze operator overloading, runtime polymorphism, Generic Programming.
6. Analyze and explore various Stream classes, I/O operations and exception handling.

|  |                              |                             |              |
|--|------------------------------|-----------------------------|--------------|
| <b>B. Sc (Information Technology)</b>                              |                              | <b>Semester – II</b>        |              |
| <b>Course Name: Object Oriented Programming with C++ Practical</b> |                              | <b>Course Code: USIT2P1</b> |              |
| <b>Periods per week (1 Period is 50 minutes)</b>                   |                              | <b>3</b>                    |              |
| <b>Credits</b>   |                              | <b>2</b>                    |              |
|  |                              | <b>Hours</b>                | <b>Marks</b> |
| <b>Evaluation System</b>   | <b>Practical Examination</b> | <b>2½</b>                   | <b>50</b>    |
|  | <b>Internal</b>              | <b>--</b>                   | <b>--</b>    |

## Course Objectives:

- The student should be able to explain the important characteristics of the C++ programming language.
- The learner must be able to combine components of the C++ programming language to develop structured program.
- The student must demonstrate the skills essential to compile, debug, and test C++ programs correctly.

|    |  |
|----|--|
| 1. |  |
| a. | Write a C++ program to create a simple calculator.   |
| b. | Write a C++ program to convert seconds into hours, minutes and seconds.  |
| c. | Write a C++ program to find the volume of a square, cone, and rectangle.   |
|    |  |
| 2. |  |
| a. | Write a C++ program to find the greatest of three numbers.   |
| b. | Write a C++ program to find the sum of even and odd n natural numbers  |
| c. | Write a C++ program to generate all the prime numbers between 1 and n, where n is a value supplied by the user.  |
| 3. |  |
| a. | Write a C++ program using classes and object Student to print name of the student, roll_no. Display the same.  |
| b. | Write a C++ program for Structure bank employee to print name of the employee, account_no. & balance. Display the same also display the balance after withdraw and deposit |
| c. | Write a C++ Program to design a class having static member function named showcount() which has the property of displaying the number of objects created of the class.     |

## Annexure I

|    |  |
|----|--|
| d. | Write a Program to find Maximum out of Two Numbers using friend function.<br>Note: Here one number is a member of one class and the other number is member of some other class.  |
| e. | Write a C++ Program using copy constructor to copy data of an object to another object.  |
| f. | Write a C++ Program to allocate memory dynamically for an object of a given class using class's constructor.   |
| 4. |  |
| a. | Write a C++ program to design a class representing complex numbers and having the functionality of performing addition & multiplication of two complex numbers using operator overloading.   |
| b. | Write a C++ program to overload new/delete operators in a class.   |
| c. | Write a C++ program to access members of a STUDENT class using pointer to object members   |
|    | Write a C++ Program to generate Fibonacci Series by using Constructor to initialize the Data Members.  |
| d. | Write a C++ Program to generate Fibonacci Series by using Constructor to initialize the Data Members.  |
| e. | Write a C++ Program that illustrate single inheritance.  |
| f. | Write a C++ Program that illustrate multiple inheritance.  |
| g. | Write a C++ Program that illustrate multi level inheritance.   |
| h. | Write a C++ Program that illustrate Hierarchical inheritance.  |
| i. | Write a C++ Program illustrating how the constructors are implemented and the order in which they are called when the classes are inherited. Use three classes named alpha, beta, gamma such that alpha, beta are base class and gamma is derived class inheriting alpha & beta  |
| 5. |  |
| a. | Write a C++ Program to design a student class representing student roll no. and a test class (derived class of student) representing the scores of the student in various subjects and sports class representing the score in sports. The sports and test class should be inherited by a result class having the functionality to add the scores and display the final result for a student. |
| 6. |  |
| a. | Write a C++ program to maintain the records of person with details (Name and Age) and find the eldest among them. The program must use this pointer to return the result.  |
| 7. |  |
| a. | Write a C++ program illustrating the use of virtual functions in class.  |
| b. | Write a C++ program to design a class representing the information regarding digital library (books, tape: book & tape should be separate classes having the base class as media). The class should have the functionality for adding new item, issuing, deposit etc. the program should use the runtime polymorphism.   |
| 8. |  |
| a. | Write a C++ program to show conversion from string to int and vice-versa.  |
| b. | Write a C++ program implementing basic operation of class ios i.e. setf, unsetf, precision etc.  |
| c. | Write a C++ program to implement I/O operations on characters. I/O operations includes inputting a string, Calculating length of the string, Storing the String in a file, fetching the stored characters from it, etc.  |
| d. | Write a C++ program to copy the contents of one file to another.   |

## Annexure I

|     |  |
|-----|--|
| e.  | Write a C++ program to perform read/write binary I/O operation on a file (i.e. write the object of a structure/class to file). |
| 9.  |  |
| a.  | Write a C++ program to implement the exception handling with multiple catch statements.  |
| b.  | Write a C++ program to implement the exception handling with rethrowing in Exception.  |
| 10. |  |
| a.  | Write a C++ Program to create Simple calculator using Class template.  |
| b.  | Write a C++ Program to get maximum of two number using Class template.   |

### Course Outcomes:

#### Leaners will able to,

- **Utilize C++ characteristics in software design and development.**
- **Explain object-oriented techniques and explain how C++ supports them.**
- **Employ C++ to demonstrate practical skill developing object-oriented solutions.**
- **Examine a problem statements and design and develop object-oriented software using good coding practices and procedures.**
- **In object-oriented design, use common software patterns and recognize their relevance in other software development contexts.**

## Annexure I

|  |                           |                             |              |
|--|---------------------------|-----------------------------|--------------|
| <b>B. Sc (Information Technology)</b>                                    |                           | <b>Semester – II</b>        |              |
| <b>Course Name: Fundamentals of Micro Processor and Microcontrollers</b> |                           | <b>Course Code: USIT202</b> |              |
| <b>Periods per week (1 Period is 50 minutes)</b>                         |                           | <b>5</b>                    |              |
| <b>Credits</b>   |                           | <b>2</b>                    |              |
|  |                           | <b>Hours</b>                | <b>Marks</b> |
| <b>Evaluation System</b>   | <b>Theory Examination</b> | <b>2</b>                    | <b>75</b>    |
|  | <b>Internal</b>           | <b>--</b>                   | <b>25</b>    |

- Course Objectives:**
- 1) To understand the basic concept of Micro Computer Systems
  - 2) To develop background knowledge in 8085 Microprocessor
  - 3) To write Assembly language Programs of 8085
  - 4) To understand the peripheral devices and interfacing to 8051 Micro Controller and design aspects of Micro Controller

| Unit       | Details   | Lectures  |
|------------|---|-----------|
| <b>I</b>   | <p><b>Microprocessor, microcomputers, and Assembly Language:</b><br/>Microprocessor, Microprocessor Instruction Set and Computer Languages, From Large Computers to Single-Chip Microcontrollers, Applications.</p> <p><b>Microprocessor Architecture and Microcomputer System:</b> Microprocessor Architecture and its operation's, Memory, I/O Devices, Microcomputer System, Logic Devices and Interfacing, Microprocessor-Based System Application.</p> <p><b>8085 Microprocessor Architecture and Memory Interface:</b> Introduction, 8085 Microprocessor unit, 8085-Based Microcomputer, Memory Interfacing, Interfacing the 8085 Memory Segment.</p>   | <b>12</b> |
| <b>II</b>  | <p><b>Interfacing of I/O Devices</b><br/>Basic Interfacing concepts, Interfacing Output Displays, Interfacing Input Devices, Memory Mapped I/O, Testing and Troubleshooting I/O Interfacing Circuits.</p> <p><b>Introduction to 8085 Assembly Language Programming:</b><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.</p> <p><b>Introduction to 8085 Instructions:</b><br/>Data Transfer Operations, Arithmetic Operations, Logic Operation, Branch Operation, Writing Assembly Languages Programs, Debugging a Program.</p>   | <b>12</b> |
| <b>III</b> | <p><b>Programming Techniques With Additional Instructions:</b><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.</p> <p><b>Counters and Time Delays:</b><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.</p> <p><b>Stacks and Sub-Routines:</b><br/>Stack, Subroutine, Restart, Conditional Call, Return Instructions, Advanced Subroutine concepts.</p> <p><b>Interrupts:</b><br/>The 8085 Interrupt, 8085 Vectored and Non vectored Interrupts, Restart as S/W Instructions.</p> | <b>12</b> |

## Annexure I

|           |   |           |
|-----------|---|-----------|
| <b>IV</b> | <p><b>Micro Controllers:</b> Embedded Systems and general purpose computer systems, history, classifications, applications and purpose of embedded systems.</p> <p><b>Embedded Hardware:</b> Memory map, i/o map, interrupt map, processor family, external peripherals, memory – RAM , ROM, types of RAM and ROM, memory testing, CRC ,Flash memory.</p> <p><b>Peripherals:</b> Control and Status Registers, Device Driver, Timer watch Timer</p> <p><b>The 8051 Microcontrollers:</b> Microcontrollers and Embedded processors, Overview of 8051 family.8051 Microcontroller hardware, Input/output pins, Ports, and Circuits, External Memory.</p> <p><b>8051 Programming in C:</b><br/>Data Types and time delay in 8051 C, I/O Programming, Logic operations, Data conversion Programs.</p> | <b>12</b> |
| <b>V</b>  | <p><b>Designing Embedded System with 8051 Microcontroller:</b>Factors to be considered in selecting a controller, why 8051 Microcontroller, Designing with 8051.</p> <p><b>Programming embedded systems:</b> structure of embedded program, infinite loop, compiling, linking and debugging.</p> <p><b>Design and Development:</b> Embedded system, development Environment – IDE, types of file generated on cross compilation, Embedded Product Development cycle and Trends in embedded Industry</p>   | <b>12</b> |

| <b>Books and References:</b> |   |                    |                  |                 |      |
|------------------------------|---|--------------------|------------------|-----------------|------|
| Sr. No.                      | Title   | Author/s           | Publisher        | Edition         | Year |
| 1.                           | Microprocessors Architecture, Programming and Applications with the 8085. | Ramesh Gaonkar     | PENRAM           | 5 <sup>th</sup> | 2012 |
| 2.                           | 8080A/8085 Assembly Language Programming                                  | Lance A. Leventhel | Osborne          |                 | 1978 |
| 3                            | Embedded Systems  | Rajkamal           | Tata Mcgraw-Hill |                 |      |
| 4                            | Introduction to embedded systems  | Shibu K V          | Tata Mcgraw-Hill | 1 <sup>st</sup> | 2012 |

### Course Outcomes:

Learners will be able to,

1. Understand the basic concepts of Micro Computer Systems
2. Understand the architecture and hardware aspects of 8085
3. Write assembly language programs in 8085
4. Design elementary aspects of Micro Controller based systems
5. Interfacing peripherals using Micro Controller



## Annexure I

|  |                              |                         |              |
|--|------------------------------|-------------------------|--------------|
| <b>B. Sc (Information Technology)</b>  |                              | <b>Semester – II</b>    |              |
| Course Name: Fundamentals of Micro Processor and Microcontrollers<br>Practical |                              | Course Code:<br>USIT2P2 |              |
| Periods per week (1 Period is 50 minutes)                                      |                              | 3                       |              |
| Credits  |                              | 2                       |              |
|  |                              | <b>Hours</b>            | <b>Marks</b> |
| <b>Evaluation System</b>   | <b>Practical Examination</b> | 2½                      | 50           |
|  | <b>Internal</b>              | --                      | --           |

**Course Objectives: The course aims to learn**

1. Operations related to single & Multiple memory locations
2. Simple assembly language programs
3. How to perform register operations, packing and unpacking
4. Embedding computer using 8051 microcontrollers
5. Interfacing I/O Ports
6. Implement flashmagic in microcontrollers

| List of Practical |  |
|-------------------|--|
| <b>1.</b>         | <b>Perform the following Operations related to memory locations.</b>   |
| a.                | Store the data byte 32H into memory location 4000H.  |
| b.                | Exchange the contents of memory locations 2000H and 4000H  |
|                   |  |
| <b>2.</b>         | <b>Simple assembly language programs.</b>  |
| a.                | Subtract two 8-bit numbers.  |
| b.                | 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. |
| c.                | Find the 1's complement of the number stored at memory location 4400H and store the complemented number at memory location 4300H.  |
| d.                | Find the 2's complement of the number stored at memory location 4200H and store the complemented number at memory location 4300H.  |
|                   |  |
| <b>3.</b>         | <b>Packing and unpacking operations.</b>   |
| 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.   |
|                   |  |
| <b>4.</b>         | <b>Register Operations.</b>  |
| 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  |

## Annexure I

|            |  |
|------------|--|
| c.         | Write a set of instructions to alter the contents of flag register in 8085.  |
| d.         | Write a program to count number of 1's in the contents of D register and store the count in the B register.  |
| <b>5.</b>  | <b>Multiple memory locations.</b>  |
| 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.   |
| <b>6.</b>  | <b>Calculations with respect to memory locations.</b>  |
| a.         | Write a program to sort given 10 numbers from memory location 2200H in the ascending order.  |
| b.         | Calculate the sum of series of even numbers from the list of numbers. The length of the list is in memory location 2200H and the series itself begins from memory location 2201H. Assume the sum to be 8 bit number so you can ignore carries and store the sum at memory location 2300H.<br>Sample problem:                                     |
| <b>7.</b>  | <b>Assembly programs on memory locations</b>   |
| a.         | 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  |
| b.         | Write an assembly language program to generate fibonacci number.   |
| c.         | Program to calculate the factorial of a number between 0 to 8.   |
| <b>8.</b>  | Design and develop a reprogrammable embedded computer using 8051 microcontrollers and to show the following aspects.<br>a. Programming<br>b. Execution<br>c. Debugging   |
| <b>9 a</b> | Configure timer control registers of 8051 and develop a program to generate given time delay.  |
| <b>b</b>   | Port I / O: Use one of the four ports of 8051 for O/P interfaced to eight LED's. Simulate binary counter (8 bit) on LED's  |
| <b>c</b>   | To interface 8 LEDs at Input-output port and create different patterns.  |
| <b>d</b>   | To demonstrate timer working in timer mode and blink LED without using any   |

## Annexure I

|            |  |
|------------|--|
|            | loop delay routine.  |
| <b>10.</b> | <b>Using FlashMagic</b>  |
| <b>a.</b>  | To demonstrate the procedure for flash programming for reprogrammable embedded system board using Flash Magic  |
| <b>b.</b>  | To demonstrate the procedure and connections for multiple controllers programming of same type of controller with same source code in one go, using flash magic. |

### Course Outcomes:

Learners will be able to,

1. Apply concepts of 8085 to single & Multiple Memory Locations
2. Apply concepts of micro-processor register operations
3. Can implement assembly language programs
4. Use of Shift registers 8 & 16 bits
5. Apply the knowledge of Flash Magic in embedded Controllers
6. Learns to simulate and configure different timer controls

## Annexure I

|   |                           |                      |              |
|---|---------------------------|----------------------|--------------|
| <b>B. Sc (Information Technology)</b>     |                           | <b>Semester – II</b> |              |
| Course Name: Web Applications Development |                           | Course Code: USIT203 |              |
| Periods per week (1 Period is 50 minutes) |                           | 5                    |              |
| Credits                                   |                           | 2                    |              |
|   |                           | <b>Hours</b>         | <b>Marks</b> |
| <b>Evaluation System</b>                  | <b>Theory Examination</b> | 2                    | 75           |
|   | <b>Internal</b>           | --                   | 25           |

**Course Objectives:**

- Understand basic concepts of Internet and World Wide Web.
- Comprehend different HTML elements that can be used to develop static web pages.
- Become familiar with concept of stylesheets and various CSS effects.
- Peruse JavaScript as a tool to add dynamism to static HTML pages.
- Explore how server-side script works on the web.
- Learn how PHP can be connected to a database to store and retrieve data.

| Unit       | Details  | Lectures  |
|------------|--|-----------|
| <b>I</b>   | <p><b>Internet and the World Wide Web:</b><br/>           What is Internet? Applications of Internet, E-mail, Telnet, FTP, E-commerce and 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, Common Features of Browsers, Search Engine, Web Server, HTTP Protocol.</p> <p><b>HTML5:</b><br/>           Introduction, Formatting Text by using Tags, Using Lists, Creating Hyperlinks and Bookmarks, Defining Metadata about an HTML Document, Redirecting to another URL.</p> <p><b>CSS:</b><br/>           Implementing Styles using CSS – Stylesheets, Formatting Text and Links using CSS, CSS Selectors, Changing Background, Adding Border, Margin and Padding, Setting Dimensions, Using Inline Container to mark up a part of a text.</p> | <b>12</b> |
| <b>II</b>  | <p><b>HTML Page Layout:</b><br/>           Using Layout Elements, Semantic Elements, Creating, Positioning and Formatting Divisions, Floating Divisions next to each other, Responsive Web Design, Inline Frames.</p> <p><b>HTML Media, Tables and Forms:</b><br/>           Embedding Images, Creating Client-side and Server-side Image Map, adding Favicon, Embedding audio and video on web page.</p> <p>Creating Simple Table, Table Dimension, Merging Table Cells, Formatting Tables: Applying Borders, Background and Foreground fills, Changing Cell Padding, Spacing and Alignment</p> <p>Collecting user input with HTML Forms, Additional Input Types in HTML5.</p>  | <b>12</b> |
| <b>III</b> | <p><b>JavaScript:</b><br/>           Introduction, Difference between Client-side and Server-side Scripting, JavaScript Variables and Constants, Data Types, Statements, Comments, Functions, Variable Scope, Hoisting, Strict Mode, JavaScript Objects, Dialog Boxes, void Keyword</p>  | <b>12</b> |

## Annexure I

|           |  |           |
|-----------|--|-----------|
|           | <p><b>Operators:</b> Arithmetic Operators, Assignment Operators, Comparison Operators, Logical Operators, Bitwise Operators</p> <p><b>Statements:</b><br/>Conditional Statements – if else, switch, Loops – while, do while, for, for in, for of, Loop Control – break, continue, labels</p> <p><b>JavaScript Objects:</b> User-defined Objects, with Keyword, Native Objects – Array, String, Date, Math, Number, RegExp</p> <p><b>DOM:</b> Introduction, DOM Properties and Methods.</p> <p><b>Browser BOM:</b> Moving back and forward with History, Cookies</p> <p><b>Events and Event Handlers:</b> HTML Events, DOM Events, DOM Event Listener, onAbort, onBlur, onChange, onClick, onDbClick, onError, onFocus, onKeyDown, onKeyPress, onKeyUp, onLoad, onMouseDown, onMouseMove, onMouseOut, onMouseOver, onMouseUp, onReset, onResize, onSelect, onSubmit, onUnload</p> |           |
| <b>IV</b> | <p><b>PHP:</b><br/>Introduction, Server-side Scripting, PHP Syntax and Comments, Variables and Constants, Data Types, Control Structures, Looping, Loop Termination, Functions, PHP Form Handling, PHP Form Validation, Superglobals, PHP Arrays, PHP Strings, PHP RegEx, PHP Numbers, PHP Math, Basic PHP Errors</p>  | <b>12</b> |
| <b>V</b>  | <p><b>Advanced PHP:</b><br/>PHP Date and Time, PHP Include, PHP Cookies, PHP Sessions, Validating and Sanitizing Data with PHP Filters</p> <p><b>PHP and MySQL</b><br/>Why PHP and MySQL? Connect to MySQL, Creating Database and Tables, Inserting Single and Multiple Rows, Retrieving Last ID, MySQL Prepared, Selecting Data, Updating Data, Deleting Data, Limiting Data.</p>   | <b>12</b> |

**Course Outcomes:**

- **Analyze working of Internet.**
- **Gain an insight into designing web pages.**
- **Use different ways of styling web pages using CSS.**
- **Implement basic and complex functionalities of JavaScript in a web page.**
- **Employ PHP Scripts to execute dynamic tasks in a web page.**
- **Perform various database tasks using PHP.**

| <b>Books and References:</b> |   |                           |                 |                 |      |
|------------------------------|---|---------------------------|-----------------|-----------------|------|
| Sr. No.                      | Title   | Author/s                  | Publisher       | Edition         | Year |
|                              | The Complete Reference HTML & CSS   | Thomas A. Powell          | McGrawHill      | 5 <sup>th</sup> | -    |
|                              | Step by Step HTML5  | Faithe Wempen             | Microsoft Press |                 | 2011 |
|                              | Learning PHP, MySQL, JavaScript, CSS & HTML5                                      | Robin Nixon               | O'Reilly        | 3 <sup>rd</sup> | 2018 |
|                              | Learning Web Design A Beginner's Guide to Html, CSS, JavaScript, And Web Graphics | Jennifer Niederst Robbins | O'Reilly        | 5 <sup>th</sup> | 2018 |

## Annexure I

|  |                                   |                                    |            |                 |      |
|--|-----------------------------------|------------------------------------|------------|-----------------|------|
|  | The Complete Reference JavaScript | Thomas A. Powell & Fritz Schneider | McGrawHill | 3 <sup>rd</sup> | 2012 |
|  | PHP & MySQL Novice to Ninja       | Tom Butler                         | SPD        | 7 <sup>th</sup> | 2022 |

|   |                              |                             |              |
|---|------------------------------|-----------------------------|--------------|
| <b>B. Sc (Information Technology)</b>                     |                              | <b>Semester – II</b>        |              |
| <b>Course Name: Web Application Development Practical</b> |                              | <b>Course Code: USIT2P3</b> |              |
| <b>Periods per week (1 Period is 50 minutes)</b>          |                              | <b>3</b>                    |              |
| <b>Credits</b>  |                              | <b>2</b>                    |              |
|   |                              | <b>Hours</b>                | <b>Marks</b> |
| <b>Evaluation System</b>                                  | <b>Practical Examination</b> | <b>2½</b>                   | <b>50</b>    |
|   | <b>Internal</b>              | <b>--</b>                   | <b>--</b>    |

### Course Objectives:

- Understand how to effectively implement HTML.
- Write CSS effectively to create well organized, styled web pages.
- Add versatility to a web page with client-side scripting.
- Deploy a local web server and run a simple web application.
- Read and process data in MySQL using PHP.

|                           |   |
|---------------------------|---|
| <b>List of Practical:</b> |   |
| <b>1.</b>                 | <b>Use of Basic Tags</b>  |
| 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 that automatically redirects the user to another page.                |
| <b>2.</b>                 | <b>Use of CSS</b>   |
| a.                        | Design a web page demonstrating different stylesheet types.                             |
| b.                        | Design a web page demonstrating grouping selectors.                                     |
| <b>3.</b>                 | <b>Layout and Media</b>   |
| a.                        | Design a web page demonstrating different semantics.                                    |
| b.                        | Design a web page embedding image, audio and video.                                     |
| c.                        | Design a web page with Imagemaps.   |
| <b>4.</b>                 | <b>Tables and Forms</b>   |
| a.                        | Design a web page with different tables.  |
| b.                        | Design a web page with a form that uses all types of controls.                          |

## Annexure I

|            |   |
|------------|---|
| <b>5.</b>  | <b>JavaScript</b>   |
| a.         | Using JavaScript, design a web page to accept a number from the user and print its Factorial.   |
| b.         | Using JavaScript, a web page that prints Fibonacci series/any given series.   |
| 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.  |
| <b>6.</b>  | <b>JavaScript Objects</b>   |
| a.         | Using JavaScript, design a web page demonstrating different native objects of JavaScript.   |
| b.         | 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). |
| <b>7.</b>  | <b>JavaScript Events</b>  |
| a.         | Write a JavaScript program to design simple calculator.   |
| b.         | Design a form and validate all the controls placed on the form using JavaScript.  |
| <b>8.</b>  | <b>Basic PHP</b>  |
| a.         | Write a PHP code to find the greater of 2 numbers. Accept the no. from the user.  |
| b.         | Write a PHP Program to accept a number from the user and print it factorial.  |
| c.         | Write a PHP program to accept a number from the user and print whether it is prime or not.  |
| d.         | Write a PHP program to display the following Binary Pyramid:<br>1<br>0    1<br>1    0    1<br>0    1    0    1<br>1    0    1    0    1 |
| e.         | Write a PHP program to demonstrate different string functions.  |
| f.         | Write a PHP program to demonstrate different array functions.   |
| <b>9.</b>  | <b>Advanced PHP</b>   |
| a.         | Write a PHP program to demonstrate use of sessions and cookies.   |
| b.         | Write a PHP program to demonstrate use of filters.  |
| <b>10.</b> | <b>PHP and MySQL</b>  |
| a.         | Write a PHP program to create:<br>Create a database College   |

## Annexure I

|    |   |
|----|---|
|    | 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. | Write a PHP program to<br>Update rows in a table<br>Delete rows from a table  |
| d. | Design a PHP page for authenticating a user   |

### Course Outcomes:

Learners will be able to,

1. Design static web pages using Hyper Text Markup Language (HTML).
2. Enhance the look of web pages by implementing CSS.
3. Collect information from the user with HTML Forms.
4. Design interactive webpages using client-side script (JavaScript).
5. Implement Document Object Model and events in web pages using JavaScript.
6. Write and deploy basic PHP code to simplify web development.
7. Store and retrieve data from a server using PHP.



## Annexure I

|  |                           |                             |              |
|--|---------------------------|-----------------------------|--------------|
| <b>B. Sc (Information Technology)</b>            |                           | <b>Semester – I</b>         |              |
| <b>Course Name: Numerical Methods</b>            |                           | <b>Course Code: USIT204</b> |              |
| <b>Periods per week (1 Period is 50 minutes)</b> |                           | <b>5</b>                    |              |
| <b>Credits</b>                                   |                           | <b>2</b>                    |              |
|  |                           | <b>Hours</b>                | <b>Marks</b> |
| <b>Evaluation System</b>                         | <b>Theory Examination</b> | <b>2</b>                    | <b>75</b>    |
|  | <b>Internal</b>           | <b>--</b>                   | <b>25</b>    |

**Course Objectives:**

Course will enhance the problem solving skills of students using extremely powerful numerical methods.

| Unit       | Details  | Lectures  |
|------------|--|-----------|
| <b>I</b>   | <p><b>Mathematical Modeling and Engineering Problem Solving:</b> A Simple Mathematical Model, Conservation Laws and Engineering Problems</p> <p><b>Approximations and Round-Off Errors:</b> Significant Figures, Accuracy and Precision, Error Definitions, Round-Off Errors</p> <p><b>Truncation Errors and the Taylor Series:</b> The Taylor Series, Error Propagation, Total Numerical Errors, Formulation Errors and Data Uncertainty</p>  | <b>12</b> |
| <b>II</b>  | <p><b>Solutions of Algebraic and Transcendental Equations:</b> The Bisection Method, The Newton-Raphson Method, The Regula-falsi method, The Secant Method.</p> <p><b>Interpolation:</b> Forward Difference, Backward Difference, Newton's Forward Difference Interpolation, Newton's Backward Difference Interpolation, Lagrange's Interpolation</p>  | <b>12</b> |
| <b>III</b> | <p><b>Solution of simultaneous algebraic equations (linear) using iterative methods:</b> Gauss-Jordan Method, Gauss-Seidel Method.</p> <p><b>Numerical differentiation and Integration:</b> Numerical differentiation, Numerical integration using Trapezoidal Rule, Simpson's 1/3rd and 3/8th rules.</p>  | <b>12</b> |
| <b>IV</b>  | <p><b>Numerical solution of 1st and 2nd order differential equations:</b> Taylor series, Euler's Method, Modified Euler's Method, Runge-Kutta Method for 1st and 2nd Order Differential Equations.</p> <p><b>Least-Squares Regression:</b> Linear Regression, Polynomial Regression, Multiple Linear Regression, General Linear Least Squares, Nonlinear Regression</p>  | <b>12</b> |
| <b>V</b>   | <p><b>Linear Programming:</b> Linear optimization problem, Formulation and Graphical solution, Basic solution and Feasible solution.</p> <p><b>Numerical Solutions of Partial Differential Equations:</b> Classification of Partial Differential Equations of the second order, Difference equation corresponding to Laplace Equation, Liebmann's Iteration Process, Bender-Schmidt's Difference Equation corresponding to the parabolic equation, Crank Nicolson's difference equations corresponding to the parabolic equation, Difference equation corresponding to the Hyperbolic equation..</p> | <b>12</b> |

**Books and References:**

| Sr. No. | Title | Author/s | Publisher | Edition | Year |
|---------|-------|----------|-----------|---------|------|
|---------|-------|----------|-----------|---------|------|

## Annexure I

|    |   |                                      |                   |                 |      |
|----|---|--------------------------------------|-------------------|-----------------|------|
| 1. | Introductory Methods of Numerical Methods | S. S. Sastry                         | PHI               | 5 <sup>th</sup> | 2012 |
| 2. | Numerical Methods for Engineers           | Steven C. Chapra, Raymond P. Canale  | Tata Mc Graw Hill | 6 <sup>th</sup> | 2010 |
| 3. | Numerical Analysis                        | Richard L. Burden, J. Douglas Faires | Cengage Learning  | 9 <sup>th</sup> | 2011 |
| 4. | Numerical Methods                         | T Veerarajan<br>T Ramachandran       | Tata Mc Graw Hill | 7 <sup>th</sup> | 2011 |

### Course Outcomes:

Learners will be able to,

1. Understand numerical techniques to find the roots of non-linear equations and solution of system of linear equations.
2. Understand the difference operators and the use of interpolation.
3. Understand numerical differentiation and integration and numerical solutions of ordinary and partial differential equations.

|  |                              |                             |              |
|--|------------------------------|-----------------------------|--------------|
| <b>B. Sc (Information Technology)</b>            |                              | <b>Semester – I</b>         |              |
| <b>Course Name: Numerical Methods Practical</b>  |                              | <b>Course Code: USIT2P4</b> |              |
| <b>Periods per week (1 Period is 50 minutes)</b> |                              | <b>3</b>                    |              |
| <b>Credits</b>                                   |                              | <b>2</b>                    |              |
|  |                              | <b>Hours</b>                | <b>Marks</b> |
| <b>Evaluation System</b>                         | <b>Practical Examination</b> | <b>2½</b>                   | <b>50</b>    |
|  | <b>Internal</b>              | <b>--</b>                   | <b>--</b>    |

### Course Objectives:

Course will provide different tools to find solutions to various numerical techniques

|                                |  |
|--------------------------------|--|
| <b>List of Practical:</b>      |  |
| To be implemented using SCILAB |  |
| <b>1.</b>                      | <b>Iterative Calculation</b>   |
| 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                          |
|                                |  |
| <b>2.</b>                      | <b>Solution of algebraic and transcendental equations:</b>               |

## Annexure I

|           |   |
|-----------|---|
| 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.                                  |
| <b>3.</b> | <b>Interpolation</b>  |
| a.        | Program for Newton's forward interpolation.   |
| b.        | Program for Newton's backward interpolation.  |
| c.        | Program for Lagrange's interpolation.   |
| <b>4.</b> | <b>Solving linear system of equations by iterative methods</b>  |
| a.        | Program for solving linear system of equations using Gauss Jordan method.   |
| b.        | Program for solving linear system of equations using Gauss Seidel method.   |
| <b>5.</b> | <b>Numerical Differentiation</b>  |
| a.        | Program to obtain derivatives numerically.  |
| <b>6.</b> | <b>Numerical Integration</b>  |
| a.        | Program for numerical integration using Trapezoidal rule.   |
| b.        | Program for numerical integration using Simpson's 1/3 <sup>rd</sup> rule.   |
| c.        | Program for numerical integration using Simpson's 3/8 <sup>th</sup> rule.   |
| <b>7.</b> | <b>Solution of differential equations</b>   |
| 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 2 <sup>nd</sup> order and 4 <sup>th</sup> order methods. |
| <b>8.</b> | <b>Regression 1</b>   |
| a.        | Program for Linear regression.  |
| b.        | Program for Polynomial Regression.  |
| <b>9.</b> | <b>Regression 2</b>   |

## Annexure I

|            |  |
|------------|--|
| a.         | Program for multiple linear regression.              |
| b.         | Program for non-linear regression.                   |
| <b>10.</b> | Numerical solution of partial differential equations |
| a.         | Program to find solution of Laplace's equation.      |

### Course Outcomes:

Learners will be able to,

1. Find fast and accurate solution to simple and complex numerical problems using these programs.

## Annexure I

|  |                           |                             |              |
|--|---------------------------|-----------------------------|--------------|
| <b>B. Sc (Information Technology)</b>            |                           | <b>Semester – II</b>        |              |
| <b>Course Name: Green IT</b>                     |                           | <b>Course Code: USIT205</b> |              |
| <b>Periods per week (1 Period is 50 minutes)</b> |                           | <b>5</b>                    |              |
| <b>Credits</b>                                   |                           | <b>2</b>                    |              |
|  |                           | <b>Hours</b>                | <b>Marks</b> |
| <b>Evaluation System</b>                         | <b>Theory Examination</b> | <b>2</b>                    | <b>75</b>    |
|  | <b>Internal</b>           | <b>--</b>                   | <b>25</b>    |

### Course Objectives:

- **To understand the concept of Green Technology.**
- **To learn Green IT regulating Green IT and different standards.**
- **To understand the concept of minimizing power utilization in technology.**
- **To know about Green PCs, Green notebooks and servers and Green data centers.**
- **To know how the way of work is changing and understand implementation of Paperless work.**
- **To know the concept of Recycling.**
- **To understand Metrics for Green IT.**

| Unit       | Details  | Lectures  |
|------------|--|-----------|
| <b>I</b>   | <p><b>Overview to Green IT:</b><br/>Problems: Toxins, Power Consumption, Equipment Disposal, Company’s Carbon Footprint: Measuring, Details, reasons to bother, Plan for the Future, Cost Savings: Hardware, Power.</p> <p><b>Regulating Green IT: Laws, Standards and Protocols</b><br/>Introduction, The Regulatory Environment and IT Manufacturers RoHS, REACH, WEEE, Legislating for GHG Emissions and Energy Use of IT Equipment. Nonregulatory Government Initiatives, Industry Associations and Standards Bodies, Green Building Standards, Green Data Centres, Social Movements and Greenpeace.</p>   | <b>12</b> |
| <b>II</b>  | <p><b>Minimizing Power Usage:</b><br/>Power Problems, Monitoring Power Usage, Servers, Low-Cost Options, Reducing Power Use, Data De-Duplication, Virtualization, Management, Bigger Drives, Involving the Utility Company, Low Power Computers, PCs, Linux, Components, Servers, Computer Settings, Storage, Monitors, Power Supplies, Wireless Devices, Software.</p> <p><b>Cooling:</b><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.</p>   | <b>12</b> |
| <b>III</b> | <p><b>Greening IT:</b><br/>Green PCs, Notebooks and Servers, Green Data Centres, Green Cloud Computing, Green Data Storage, Green Software, Green Networking and Communications.</p> <p><b>Changing the Way of Work:</b><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.</p> <p><b>Going Paperless:</b><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, Microsoft Office SharePoint Server 2007, Electronic Data Interchange (EDI), Nuts and Bolts, Value Added Networks, Advantages, Obstacles.</p> | <b>12</b> |

## Annexure I

|           |   |           |
|-----------|---|-----------|
| <b>IV</b> | <p><b>Recycling:</b><br/>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.</p> <p><b>Hardware Considerations:</b><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, Using Remote Desktop, Establishing a Connection.</p> | <b>12</b> |
| <b>V</b>  | <p><b>Greening Your Information Systems:</b><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.</p> <p><b>Staying Green:</b><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.</p>   | <b>12</b> |

| <b>Books and References:</b> |   |  |                                    |         |      |
|------------------------------|---|--|------------------------------------|---------|------|
| 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.                      |         |      |
| 3.                           | Green Data Center: Steps for the Journey                                    | Alvin Galea, Michael Schaefer, Mike Ebbers   | Shroff Publishers and Distributers |         | 2011 |
| 4.                           | Green IT  | Deepak Shikarpur                             | Vishwkarma Publications,           |         | 2014 |
| 5.                           | Green Computing Tools and Techniques for Saving Energy, Money and Resources | Bud E. Smith                                 | CRC Press                          |         | 2014 |
|                              | Green Computing and Green IT Best Practice                                  | Jason Harris                                 | Emereo                             |         |      |

### Course Outcomes:

#### Learners will be able to,

- Understand the concept of Green IT and problems related to it.
- Know different standards for Green IT.
- Understand the how power usage can be minimized in Technology.
- Learn about how the way of work is changing.
- Understand the concept of recycling.
- Know how information system can stay Green Information system.

## Annexure I

|   |                              |                      |              |
|---|------------------------------|----------------------|--------------|
| <b>B. Sc (Information Technology)</b>     |                              | <b>Semester – II</b> |              |
| Course Name: Practical's in PL/SQL        |                              | Course Code: USIT2P5 |              |
| Periods per week (1 Period is 50 minutes) |                              | 3                    |              |
| Credits                                   |                              | 2                    |              |
|   |                              | <b>Hours</b>         | <b>Marks</b> |
| <b>Evaluation System</b>                  | <b>Practical Examination</b> | 2½                   | 50           |
|   | <b>Internal</b>              | --                   | --           |

**Course Objectives:**

- To understand the basics of PL/SQL.
- To understand control and conditional statement in PL/SQL.
- To understand working of sequences and cursor in PL/SQL.
- To understand concept of stored procedure and functions.
- To understand triggers and packages in PL/SQL.
- To understand the concept of Exception handling.

| <b>List of Practical:</b> |  |
|---------------------------|--|
| <b>1.</b>                 | <b>PL/SQL Basics</b>                             |
| a.                        | Use of variables.                                |
| b.                        | Write executable statement.                      |
| c.                        | Interacting with Oracle Server.                  |
| d.                        | Create anonymous PL/SQL block                    |
| <b>2.</b>                 | <b>Control Structure in PL/SQL .</b>             |
| a.                        | Using while loop                                 |
| b.                        | Do loop  |
| c.                        | For loop   |
| d.                        | Use of GOTO statement                            |
| <b>3.</b>                 | <b>Create conditional statement using PL/SQL</b> |
| a.                        | Using if statement                               |
| b.                        | Using if else statement                          |
| c.                        | Using elsif ladder                               |
| d.                        | Using case expression.                           |
| <b>4.</b>                 | <b>Creation of Sequence in PL/SQL</b>            |
| <b>5.</b>                 | <b>Create cursor in PL/SQL</b>                   |
| a.                        | Implicit cursor                                  |
| b.                        | Explicit   |

## Annexure I

|            |   |
|------------|---|
| c.         | Parameterized cursor                    |
| d.         | Cursor for loop                         |
| <b>6.</b>  | <b>Creation of Procedures in PL/SQL</b> |
| <b>7.</b>  | <b>Functions in PL/SQL</b>              |
| a.         | Compute and returns the maximum value   |
| b.         | Compute factorial of given number.      |
| <b>8.</b>  | <b>Creation of Trigger</b>              |
| a.         | Create Row level trigger                |
| b.         | Create Statement level trigger          |
| c.         | Create instead of trigger               |
| <b>9.</b>  | <b>Handling exceptions</b>              |
| a.         | Creation of user defined exception      |
| b.         | Creation of system defined exception.   |
| <b>10.</b> | <b>Creation of Package in PL/SQL</b>    |

### Course Outcomes:

#### Learner will be able to:

- **Understand the basics of PL/SQL.**
- **Use of the control and conditional statement in PL/SQL.**
- **Apply sequences and cursor in PL/SQL.**
- **Know the concept of stored procedure and functions**
- **Create the triggers and packages in PL/SQL.**
- **Implement the concept of Exception handling.**