SEMESTER II

Syllabus
B.Sc. (Information Technology)
(Sem.- II)

Major Courses

Name of the Course: Object Oriented Programming using C++

Sr.No.	Heading	Particulars
1	**Description the course : Including but Not limited to:**	This course provides students knowledge and skills to understand and implement the object oriented skills. It will help them to implement OOP solutions to real-world problems.
2	**Vertical :**	Major
3	**Type :**	Theory
4	**Credits :**	2 credits (1 credit = 15 Hours for Theory in a semester)
5	**Hours Allotted :**	30 Hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives(CO):** CO 1. To explain the difference between object oriented programming and procedural programming. CO 2. To understand OOP principles to create modular, reusable, and maintainable code. CO 3. To understand the concept of polymorphism ,virtual functions,inheritance and exception handling. CO 4. To understand file handling concepts using C++.	
8	**Course Outcomes (OC):** OC 1. Students can explain the key concept of OOP and their application in software development. OC 2. Students can Design and implement classes and objects to model real-world entities. OC 3. Students can apply the concepts of polymorphism, virtual functions, inheritance and exception handling in program. OC 4. Students can apply operator overloading, runtime polymorphism, generic Programming OC 5. Students can implement file handling concepts in program	
9	**Modules:-** **Module 1:**	
	1. Object Oriented Methodology: Introduction, Advantages and Disadvantages of Procedure Oriented Languages, Application of OOPS, Principles of OOPS: Objects, Classes, Data Abstraction and Data Encapsulation, Inheritance, Polymorphism, Dynamic Binding, Message Passing. 2. Classes and Objects: Simple classes (Class specification, class members accessing), Defining member functions, passing object as an argument, Returning object from functions, friend classes, friend function.	**15 Hrs**

	3. Constructors and Destructors: Introduction, Default Constructor, Parameterized Constructor and examples, Destructors. 4. Program development using Inheritance: Introduction, Advantages provided by inheritance, choosing the access specifier, Derived class declaration, derived class constructors, class hierarchies, multiple inheritance, multilevel inheritance, hybrid inheritance.	
	Module 2:	
	5. Polymorphism: Concept of function overloading, overloaded operators, overloading unary and binary operators. 6. Virtual Functions: Introduction and need, Pure Virtual Functions, this Pointer, abstract classes, virtual destructors. 7. Exception Handling: Introduction, Exception Handling Mechanism, Concept of throw & catch with example. 8. Working with Files: Introduction, File Operations, Various File Modes, File Pointer and their Manipulation.	**15 Hrs**
10	**Text Books** 1. Object-oriented Programming C++, Hari Mohan Pandey , Laxmi Publications 2. C++ Programming: An Object-Oriented Approach, Behrouz A. Forouzan, Richard F. Gilberg , McGraw-Hill Education 3. C++ How to Program , Paul Deitel, Harvey Deitel	
11	**Reference Books** 1. Object Oriented Programming in C++ , E Balagurusamy 2. Object-Oriented Programming in C++ , Robert Lafore, Pearson Education. 3. Programming with ANSI C++ , Bhushan Trivedi 4. Demystified Object- Oriented Programming with C++, Dorothy R. Kirk	
12	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**
13	**Continuous Evaluation through:** Class test of 1 of 15 marks Class test of 2 of 15 marks Average of the two: 15 marks Quizzes/ Presentations/ Assignments: 5 marks Total: 20 marks	**Format of Question Paper:** **External Examination (30 Marks)– 1 hr duration**
14	**Format of Question Paper:** (**Semester End Examination : 30 Marks. Duration:1 hour)** Q1: Attempt any two (out of four) from Module 1 (15 marks) Q2: Attempt any two (out of four) from Module 2 (15 marks)	

Name of the Course: Web Designing

Sr.No	Heading	Particulars
1	**Description the course : Including but Not limited to:**	The objective of Web Designing course is to provide instructions on creating and maintaining a web page for publishing on the Internet. Students will be able to use HTML editor to author pages that include text and graphics..
2	**Vertical :**	Major
3	**Type :**	Theory
4	**Credits :**	2 credits (1 credit = 15 Hours for in a semester)
5	**Hours Allotted :**	30 Hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives(CO):** CO 1. To understand the fundamentals of Internet, and the principles of web design CO 2. To develop basic websites using HTML and Style Sheets. CO 3. To understand different style sheets used in web designing. CO 4. To implement JavaScript as a tool to add dynamism to static HTML pages.	
8	**Course Outcomes (OC):** OC 1. Learners will be able to use the HTML programming language OC 2. Learners will be able to execute web pages designed using HTML OC 3. Describe the concepts of World Wide Web, and the requirements of effective web design OC 4. List various tags in html and **use** these to **create** web page OC 5 : Gain necessary skills for designing and developing web applications	
9	**Modules:-** **Module 1:**	
	1. Introduction to HTML 5: What Is HTML? Understanding HTML Tags, Setting Up the Document Structure: Specifying the Document Type, Creating the HTML, Specifying a Page Title. Formatting Text by Using Tags: Creating Headings, Applying Bold and Italic Formatting, Applying Superscript and Subscript Formatting, Using Monospace and Preformatted Text. Using Lists and Backgrounds: Creating Bulleted and Numbered Lists, Creating Definition Lists, Inserting Special Characters, Inserting Horizontal Lines, Choosing Background and Foreground Colors. Creating Hyperlinks and Anchors- Hyperlinking to a Web Page, Creating Hyperlinking to an E-Mail Address, Hyperlinking to Other Content. **Style Sheets and Graphics**: Introduction to Style Sheets: Understanding Styles, Constructing Style Rules, Creating Styles for Nested Tags, Applying Styles to Hyperlinks, Creating and Linking to External Style Sheets.	**15 Hrs**

	Formatting Text by Using Style Sheets: Specifying a Font Family, Specifying a Font Size and Color, Applying Bold and Italics, Applying Strikethrough and Underlining, Creating Inline Spans, Adjusting Spacing Between Letters. Formatting Paragraphs by Using Style Sheets: Indenting Paragraphs, Applying a Border to a Paragraph, Specifying the Horizontal Alignment of a Paragraph, **Displaying Graphics** Selecting a Graphics Format, Preparing Graphics for Web Use, Inserting Graphics, Arranging Elements on the Page, Controlling Image Size and Padding, Hyperlinking from Graphics, Using Thumbnail Graphics, Including Alternate Text for Graphics, Adding Figure Captions 2. Page Layout and Navigation- Creating Navigational Aids , Creating a Text-Based and Graphical Navigation Bar, Creating an Image Map, Creating Tables, Specifying the Size of a Table, Specifying the Width of a Column, Merging Table Cells. Formatting Tables-Applying Table Borders, Applying Borders by Using Attributes, Applying Borders by Using Styles, Changing Cell Padding, Spacing, and Alignment. Setting Horizontal and Vertical Alignment **Creating User Forms**- Creating a Basic Form- Creating a Text Box,Special Field types for E-Mail and Web Addresses, Creating a Text Area, Creating a Submit or Clear Button, Creating Check Boxes and Option Buttons, Additional Input Types in HTML5 Incorporating Sound and Video- What's New with Audio and Video in HTML5?,Embedding Video Clips- Introducing the <video> Tag, The <embed> Tag: Your Fallback Plan, Placing a Video Clip on a Web Page. Incorporating Audio on a Web Page- Playing Audio with the <audio> Tag, Placing an Audio Clip on a Web Page	
	Module 2:	
	1. JavaScript: Introduction to JavaScript: Variable, statements, Operators, Comments, constructs, Functions, expressions, JavaScript console, Scope, Events, Strings, String Methods, Numbers, Number Methods, Dates, Date Formats, Date,Methods,Arrays, Array Methods, Booleans, Comparisons, Control Structures: Conditions, Switch, Loop For, Loop While, Break. **Operators**: Arithmetic Operators, Assignment Operators, Comparison Operators, Logical Operators, Bitwise Operators Statements: Conditional Statements – if else, switch, Loops – while, do while, for, for in, for of, Loop Control – break, continue, labels JavaScript Objects: User-defined Objects, with Keyword, Native Objects – Array, String, Date, Math, Number, RegExp, Cookies Events and Event Handlers: HTML Events, DOM Events, DOM Event Listener,onAbort, onBlur, onChange, onClick, onDblClick, onError, onFocus, onKeyDown,onKeyPress, onKeyUp, onLoad, onMouseDown, onMouseMove, onMouseOut,onMouseOver, onMouseUp, onReset, onResize, onSelect, onSubmit, onUnload 2. Basics of JQuery, JQuery selection and events, JQuery Effects, JQuery traversal and manipulation, Data attributes and templates, jQuery Plugins.	**15 Hrs**

	3. JSON – JSON: Introduction, JSON Grammar, JSON Values, JSON Tokens, Syntax, JSON vs. XML, Data Types, Objects, Arrays, Creating JSON, JSON Object, Parsing JSON, Persisting JSON, Data Interchange, JSON HTML, JSONP	
10	**Text Books** 1. Step by Step HTML5 by Faithe Wempen, Microsoft Press,2011 2. The Complete Reference HTML & CSS, Thomas A. Powell. McGrawHill, 5 th Edition,2010 3. The Complete Reference JavaScript Thomas A. Powell &Fritz Schneider McGrawHill 3rd 2012 4. Web Technologies: HTML, JAVASCRIPT, PHP, JAVA, JSP, XML and AJAX, Black Book Kindle Edition,by Kogent Learning Solutions Inc 5. HTML 5 Black Book, Covers CSS 3, JavaScript, XML, XHTML, AJAX, PHP and jQuery, 2ed Kindle Edition,by DT Editorial Services 6. JSON at work ,Tom MArrs,O'REILLY,First edition,2017	
11	**Reference Books** 1. Learning Web Design A Beginner's Guide to Html, CSS, JavaScript, And Web Graphics, Jennifer Niederst Robbins, O'Reilly, 5th Edition,2018. 2. Ivan Bayross, "Web Enabled Commercial Applications Development using HTML, DHTML, Javascript, Perl CGI", BPB, 2004 3. HTML 5 for Web Designers (By: Jeremy Keith) – http:// freepdf-books.com 4. Introduction to JavaScript Object Notation: A To-the-Point Guide to JSON kindle Edition by Lindsay Bassett, O'REILLY	
12	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**
13	**Continuous Evaluation through:** Class test of 1 of 15 marks Class test of 2 of 15 marks Average of the two: 15 marks Quizzes/ Presentations/ Assignments: 5 marks Total: 20 marks	**Format of Question Paper: External Examination (30 Marks)– 1 hr duration**
14	**Format of Question Paper:** (**Semester End Examination : 30 Marks. Duration:1 hour)** Q1: Attempt any two (out of four) from Module 1 (15 marks) Q2: Attempt any two (out of four) from Module 2 (15 marks)	

Name of the Course: Major Practical II

Sr.No.	Heading	Particulars
1	**Description the course :** **Including but Not limited to:**	**Object Oriented Programming usng C++ Practical** OOP encourages modular objects for reusable code, ensures well-organized and maintainable code via encapsulation, inheritance, and polymorphism, allowing flexibility and easy updates. Additionally, OOP models real-world scenarios, enhancing system understanding. **Web Designing Practical** Applying basic programming principles to the construction of websites

2	**Vertical :**	Major Practical
3	**Type :**	Practical
4	**Credits :**	2 credits (Total 60 hrs; 1 credit = 15 Hours for Theory or 30 Hours of Practical work in a semester)
5	**Hours Allotted :**	60 Hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives(CO):** CO 1. To explain the important characteristics of the C++ programming language. CO 2. To combine components of the C++ programming language to develop structured program. CO 3. To demonstrate the skills essential to compile, debug, and test C++ programs correctly. CO 4. To understand how to effectively implement HTML. CO 5. To develop the concept of basic and advanced text formatting. CO 6. To understand Hyper linking, Designing of webpage.	
8	**Course Outcomes (OC):** OC 1. Utilize C++ characteristics in software design and development. OC 2. Explain object-oriented techniques and explain how C++ supports them. OC 3. Employ C++ to demonstrate practical skill developing object-oriented solutions. OC 4. Examine a problem statements and design and develop object-oriented software using good coding practices and procedures. OC 5. Design static web pages using Hyper Text Markup Language (HTML). OC 6. Use their learned skills, knowledge and abilities to develop web sites OC 7. Collect information from the user with HTML Forms. OC 8. Enhance the look of web pages by implementing audio and video	
9	**Module I**	
	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	**30 Hrs**

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.

d. Write a program to demonstrate function definition outside class and accessing class members in function definition.

4.

a. Write a friend function for adding the two complex numbers, using a single class

b. Write a friend function for adding the two different distances and display its sum, using two classes.

c. Write a friend function for adding the two matrix from two different classes and display itssum

d. Write a Program to find Maximum out of Two Numbers using friend function.

Note: Here one number is a member of one class and the other number is member of some other class.

5.

a. Design a class Complex for adding the two complex numbers and also show the use of constructor.

b. Design a class Geometry containing the methods area() and volume() and also overload the area()function

c. Design a class StaticDemo to show the implementation of static variable and staticfunction

d. Write a C++ program to overload new/delete operators in a class.

e. Write a C++ Program to generate Fibonacci Series by using Constructor to initialize the Data Members.

6.

a. Overload the operator unary(-) for demonstrating operator overloading

b. Overload the operator + for adding the timings of two clocks, And also pass objects as an argument.

c. Overload the + for concatenating the two strings. For e.g “Py” +“thon” =Python

7.

a. Implement the concept of method overriding.

b. Show the use of virtual function

c. Show the implementation of abstract class.

8.

a. Write a C++ Program that illustrate single inheritance.

b. Write a C++ Program that illustrate multiple inheritance.

c. Write a C++ Program that illustrate multi-level inheritance.

d. Write a C++ Program that illustrate Hierarchical inheritance.

9.

a. Show the implementation of exception handling

b. Show the implementation for exception handling for strings
c. Show the implementation of exception handling for using the pointers.

10.
a. Design a class FileDemo open a file in read mode and display the total number of words and lines in the file.
b. Design a class to handle multiple files and file operations
c. Design a editor for appending and editing the files

Module II

1 Use of Basic and Advanced Tags, Lists and Backgrounds
a. Understanding elements, Tags and basic structure of HTML files
b. Design a web page using basic and advanced text formatting tags.
c. Design a web page using ordered, unordered list and description list.
d. Design a web page by choosing Background and Foreground Colors
e. Design a web page using Nested list and special characters.
f. Write an HTML code to display your CV on a web page.

2 Creating Hyperlinks, Anchors and style sheets
a. Design a web page with links to different pages and allow navigation between web pages.
b. Design a web page that automatically redirects the user to Other Content
c. Creating Hyperlinking to an E-Mail Address
d. Design a web page for creating Styles for Nested Tags
e. Design a web page by applying Styles to Hyperlinks
f. Design a web page by Creating and Linking to External Style Sheets.

3 Formatting Text and Paragraph by Using Style Sheets and displaying graphics
a. Design a web page by using text formatting tags
b. Design a web page using Indenting Paragraphs, Applying Border to a Paragraph and Specifying Horizontal Alignment of a Paragraph
c Implement a web page by creating inline spans and adjusting space between lines
d. Implement a web page by inserting a image and controlling the image size and padding
e. Design a web page by making image as a hyperlink
f. Develop a web page by using thumbnail graphics and also implement text for graphics

4 Tables , Page Layout and Navigation
a. Display a time table and display it in tabular format,
b. Write an html program to get the following output

NAME	SUBJECT	MARKS

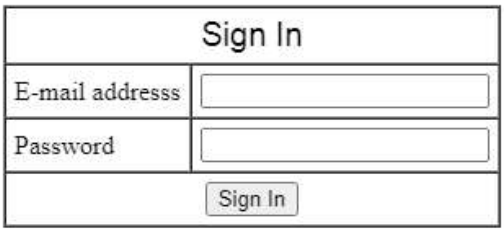
30 Hrs

Hillary	Advanced Web	75
	Operating System	60
Lary	Advanced Web	80
	Operating System	75
	Total Average: 72.5	

c. Design a table by merging the table cells.
d. Design a web page by Creating a Text-Based Navigation Bar
e . Design a web page by Creating a Graphical Navigation Bar
f. Design a web page with Image Map

5. Forms and Introducing video and audio tags

a. Design a web page with a form that uses all types of controls.
b. Design an admission form for any course in your college with text, pass word fields, check boxes, radio button and reset button.
c. Write a program to get the following output



d. Design a web page by placing a Video Clip on a Web Page
e. Design a web page by placing an Audio Clip on a Web Page
f. Design a web page embedding image, audio and video.

6 .Basics of java script
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.

7. Java Script: Validating User fields
a. Demonstrate the use of Document object methods.
b. Using java script, demonstrate validating Text Input Fields, Drop-down Lists and Checkboxes
c. Using java script, demonstrate validating Radio buttons and Validating Multi-Select Boxes
d. Write a Java script to prompt for users name and display it on the screen.

	8. Java Script : Handling the events a.Using java script, demonstrate the use of onAbort, onBlur, onChange, onClick, onDblClick events b. Using java script, demonstrate the use of onDragDrop, onError, onFocus events c. Using java script, demonstrate the use of onKeyDown, onKeyPress, onKeyUp, onLoad, onReset, onResize, onSelect, onSubmit, onUnload events d. Using java script, demonstrate the use of onMouseDown, onMouseMove, onMouseOut, onMouseOver, onMouseUp, onMove events. e. Using java script, demonstrate the use of onKeyDown, onKeyPress, onKeyUp, onLoad, onReset, onResize, onSelect, onSubmit, onUnload events 9. JQuery a. use JQuery effect in page b. Write a jQuery Code to find the data passed with the on() method for each element. c. Use JQuery Events d. JQuery traversal and manipulation 10. JSON Basics and Working with JSON a. Creating JSON b. Parsing JSON c. Persisting JSON d. Demonstrate use of JSON objects in array, print array on web page using document object e. Read data from json file and convert it into a JavaScript object and display the data in web page using document object	
10	Text Books 1. Object-oriented Programming C++, Hari Mohan Pandey 2. C++ Programming: An Object-Oriented Approach, Behrouz A. Forouzan, Richard F. Gilberg 3. C++ How to Program, Paul Deitel, Harvey Deitel 4. Step by Step HTML5, Faithe Wempen, Microsoft Press,2011 5. The Complete Reference HTML & CSS, Thomas A. Powell. McGraw Hill, 5th Edition,2010	
11	**Reference Books** 1. Object Oriented Programming in C++, E Balagurusamy 2. Object-Oriented Programming in C++ by Robert Lafore 3. Programming with ANSI C++, Bhushan Trivedi 4. Demystified Object- Oriented Programming with C++, Dorothy R. Kirk 5. Learning Web Design A Beginner's Guide to Html, CSS, JavaScript, And Web Graphics, Jennifer Niederst Robbins, O'Reilly, 5th Edition,2018. 6. "Web Enabled Commercial Applications Development using HTML, DHTML, Javascript, Perl CGI", Ivan Bayross, BPB, 2004 7. HTML 5 for Web Designers (By: Jeremy Keith) – http:// freepdf-books.com	
12	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**

13	**Continuous Evaluation through:** Students are expected to attend each practical and submit the written practical of the previous session. Performing Practical and writeup submission will be continuous internal evaluation. 2.5 marks can be awarded for each practical performance and writeup submission totalling to 50 marks and can be converted to 20 marks.	30 marks practical exam of 2 hours duration
14	**Format of Question Paper: Duration 2 hours. Certified copy of Journal is compulsory to appear for the practical examination** Practical Slip: Q1. From Module 1 13 marks Q2. From Module 2 12marks Q3. Journal and Viva 05 marks	

Vocational Skill Courses (VSC)

Name of the Course: Assembly Language Programming

Sr.No.	Heading	Particulars
1	**Description the course : Including but Not limited to:**	Introduction: The 8085 Assembly Language Programming course covers the principles and practices of writing low-level software that controls the 8085 microprocessor. This course provides an in-depth understanding of the 8085 microprocessor architecture and its instruction set, as well as how to write, debug, and optimize assembly language programs for this microprocessor. Relevance and Usefulness: The course is relevant to computer science/engineering students interested in learning about microprocessors and embedded systems programming. The course provides the fundamental knowledge and skills required to design and implement computer systems with low-level software control. Assembly language programming is the foundation of modern computer technology, which makes the course relevant to anyone interested in computer systems and programming. Application and Interest: The course is essential for students aspiring to work in the field of embedded systems, microcontroller/microprocessor programming, or any programming role that involves low-level software development. By the end of the course, students will be able to write efficient and optimized assembly language programs that control the functionality of a microprocessor. Connection with Other Courses: 8085 Assembly Language Programming is a fundamental course that provides an understanding of how computer systems work at the lowest level. It connects with several other computer science courses, such as Computer Organization and Architecture, Operating Systems, Compiler Design, and Embedded Systems Design. Demand in the Industry and Job Prospects: There is a high demand in the industry for programmers who possess knowledge of low-level software development, such as programming microprocessors with assembly language. Many industries, including aerospace, automotive, healthcare, and consumer electronics, require low-level software development skills in their employees. Job prospects for graduates with expertise in 8085 Assembly language

		programming are abundant in these sectors. Job roles may include embedded software engineer, hardware engineer, firmware developer, software developer, and testing/validation engineer.
2	**Vertical :**	Vocational Skill Course(VSC)
3	**Type :**	Practical
4	**Credits :**	2 credits (60 hours in a semester)
5	**Hours Allotted :**	60 Hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives(CO):** CO 1. To gain a thorough understanding of the 8085 microprocessor architecture and its associated instruction set. CO 2. To develop the ability to write and debug assembly language programs for the 8085 microprocessor. CO 3. To learn the principles of computer organization and how they relate to the 8085 microprocessor. CO 4. To become proficient in the use of 8085 assembly language programming tools, simulators, and debuggers. CO 5. To learn how to interface different input/output devices with the 8085 microprocessor. CO 6. To understand the concept of interrupts and how they are used in 8085 assembly language programming.	
8	**Course Outcomes(CO):** OC 1. Explain the architecture of the 8085 microprocessor and its associated instruction set. OC 2. Identify the different types of registers and their functions in the microprocessor. OC 3. Describe the memory organization and addressing modes of the 8085 microprocessor. OC 4. Write assembly language programs for the 8085 microprocessor using various instructions and addressing modes. OC 5. Debug and troubleshoot assembly language programs for the 8085 microprocessor using simulators and debuggers. OC 6. Implement conditional branching and looping constructs in assembly language programs. OC 7. Use 8085 assembly language programming tools, such as editors, assemblers, and emulators for developing and testing programs. OC 8. Simulate microprocessor operations using emulators and debuggers. OC 9. Connect input/output devices, such as LEDs, switches, and displays, to the 8085 microprocessor. OC 10.	
9	**Modules:-** **Module 1:**	
	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.**	**30 Hrs**

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 I'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 I's in the contents of D register and store the count in the B register.

5. Multiple memory locations.

a. Calculate the sum of series of numbers. The length of the series is in memory location 4200H and the series begins from memory location 4201H. a. Consider the sum to be 8 bit number. So, ignore carries. Store the sum at memory location 4300H. b. Consider the sum to be 16 bit number. Store the sum at memory 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.	
	Module 2:	
	1. Calculations with respect to memory locations. a. Write a program to sort given 10 numbers from memory location 2200H in the ascending order. b. Calculate the sum of series of even numbers from the list of numbers. The length of the list is in memory location 2200H and the series itself begins from memory location 2201H. Assume the sum to be 8 bit number so you can ignore carries and store the sum at memory location 2Sample problem: c. Calculate the sum of series of odd numbers from the list of numbers. The length of the list is in memory location 2200H and the series itself begins from memory location 2201H. Assume the sum to be 16-bit. Store the sum at memory locations 2300H and 2301H. d. Find the square of the given numbers from memory location 6100H and store the result from memory location 7000H e. Search the given byte in the list of 50 numbers stored in the consecutive memory locations and store the address of memory location in the memory locations 2200H and 2201H. Assume byte is in the C register and starting address of the list is 2000H. If byte is not found store 00 at 2200H and 2201H f. 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 I, array2 and array3 are 2200H, 2300H and 2400H, respectively **2. 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: c. 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.	**30 Hrs**

	d. 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. e. 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 f. Write an assembly language program to generate Fibonacci number. g. Program to calculate the factorial of a number between 0 to 8. **3. 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. **4. 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 **5. 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	
10	**Text Books** 1. 8080A/8085 Assembly Language Programming, Lance A. Leventhel, Osborne, 1978	

11	**Reference Books** 1. Microprocessors Architecture, Programming and Applications with the 8085, Fifth Edition, Penram Publications, 2012	
12	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**
13	**Continuous Evaluation through:** Students are expected to attend each practical and submit the written practical of the previous session. Performing Practical and writeup submission will be continuous internal evaluation. 2.5 marks can be awarded for each practical performance and writeup submission totalling to 50 marks and can be converted to 20 marks.	30 marks practical exam of 2 hours duration
14	**Format of Question Paper: Duration 2 hours. Certified copy of Journal is compulsory to appear for the practical examination** Practical Slip: Q1. From Module 1 13 marks Q2. From Module 2 12marks Q3. Journal and Viva 05 marks	

Skill Enhancement Courses (SEC)

Name of the course : Web Programming

Sr.No.	Heading	Particulars
1	**Description the course : Including but Not limited to:**	This course covers a range of topics aimed at equipping students with the skills and knowledge needed to create visually appealing, functional, and user-friendly websites. The course provides an insight into emerging technologies to design and develop state of the art web applications using client-side scripting, server-side scripting, and database connectivity. website development includes all related development tasks, such as client-side scripting, server-side scripting, server and network security configuration, eCommerce development, and content management system (CMS) development. Website design is a combination of different elements that work together to create an effective and user-friendly experience. These include the use of typography, layout, color theory, grid systems, motion graphics, and responsive designs.
2	**Vertical :**	Skill Enhancement Course(SEC)
3	**Type :**	Practical
4	**Credits:**	2 credits (1 credit = 30 Hours of Practical work in a semester)
5	**Hours Allotted :**	60 Hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives (CO)** **CO1:** To understand how to use Java script objects and XML. **CO2:** To create well organized, styled web pages **CO3:** To add versatility to a web page using jQuery **CO4:** To deploy a local web server and run a simple web application. **CO5:** To read and process data in MySQL using PHP. **CO6:** To understand usage of Bootstrap	
8	**Course Outcomes (OC)** OC1: Knowledge in different java script objects. OC2: How to use XML with CSS and XSL OC3: validate a form using jQuery OC4: handle asynchronous requests OC4: Write and deploy PHP with database and to simplify web development. OC5: Create a responsive layout using the Bootstrap	

	9	**Modules:** **Module 1:** **1. Write JavaScript code for** a. Demonstrating different JavaScript Objects such as String, RegExp, Math, Date b. Demonstrating different JavaScript Objects such as Window, Navigator, History, Location, Document c. Storing and Retrieving Cookies **2. Create a XML file with Internal / External DTD and display it using** a. CSS b. XSL **3. Write PHP scripts for- Performing certain mathematical operations such as calculating factorial / finding Fibonacci Series / Displaying Prime Numbers in a given range / Evaluating Expressions** **4. Write PHP scripts for** a. Retrieving data from HTML forms b. Working with Arrays c. Working with Files (Reading / Writing) **5. Advanced PHP** a. Write a PHP program to demonstrate use of sessions and cookies. b. Write a PHP program to demonstrate use of filters.	**30 Hrs**
		Module 2 **6. PHP and MySQL** a. Write a PHP program to create: Create a database College b. Create a table Department (Dname, Dno, Number_of_faculty) c. 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. **7. Write a PHP program** a. Update rows in a table b. Delete rows from a table **8. Design a PHP page for authenticating a user** **9. Write PHP scripts for** a. Storing and Retrieving Cookies b. Storing and Retrieving Sessions **10. Perform the following using Bootstrap:** a. Create a responsive layout using the Bootstrap grid system b. Create a simple Bootstrap navbar with dropdown menus c. Create a basic Bootstrap form with validation	**30 Hrs**
	10	**Text Books** • HTML 5 Black Book, Covers CSS 3, JavaScript, XML, XHTML, AJAX, PHP and jQuery, 2ed, Dreamtech Press, 2016 • Web Programming and Interactive Technologies, scriptDemics, StarEdu Solutions India, 2018	

	• PHP: A Beginners Guide, Vikram Vaswani, TMH	
11	**Reference Books** • HTML, XHTML, and CSS Bible Fifth Edition, Steven M. Schafer, WILEY, 2011 • Learning PHP, MySQL, JavaScript, CSS & HTML5, Robin Nixon, O'Reilly, 2018 • PHP, MySQL, JavaScript & HTML5 All-in-one for Dummies, Steve Suehring, Janet Valade Wiley, 2018	
12	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**
13	**Continuous Evaluation through:** Students are expected to attend each practical and submit the written practical of the previous session. Performing Practical and writeup submission will be continuous internal evaluation. 2.5 marks can be awarded for each practical performance and writeup submission totalling to 50 marks and can be converted to 20 marks.	30 marks practical exam of 2 hours duration
14	**Format of Question Paper: Duration 2 hours. Certified copy of Journal is compulsory to appear for the practical examination** Practical Slip: Q1. From Module 1 13 marks Q2. From Module 2 12marks Q3. Journal and Viva 05 marks	

Name of the Course: PLSQL Practical

Sr.No.	Heading	Particulars	
1	**Description the course : Including but Not limited to:**	PL/SQL ,Oracle's procedural extension language for SQL, allows developers to include procedural language components such as loops, conditional statements and functions. The course enables students with practical experience in using PL/SQL for effective database programming and development.	
2	**Vertical :**	Skill Enhancement Course(SEC)	
3	**Type :**	Practical	
4	**Credits :**	2 credits	
5	**Hours Allotted :**	60 Hours	
6	**Marks Allotted:**	50 Marks	
7	**Course Objectives(CO):** CO 1. Comprehend the basics of PL/SQL and gain knowledge about control and conditional statement in PL/SQL. CO 2. Understand working with cursors,collections and composite data types in PL/SQL. CO 3. Develop expertise in creating stored procedures and functions. CO 4. Explore the use of triggers to automate responses to events within the database. CO 5. Understand the concept of Exception handling. CO 6. Design modular applications using packages.		
8	**Course Outcomes (OC):** OC 1. Use PL/SQL variables ,data types, control and conditional statement. OC 2. Apply sequences and cursor in PL/SQL. OC 3. Work with Collection and Composite Data Types. OC 4. Develop PL/SQL structures like functions, procedures and triggers for database applications. OC 5. Handle errors and exceptions in PL/SQL programs. OC 6. Develop PL/SQL packages.		
9	**Modules:-** **Module 1:**		
	1. **PL/SQL Basics**- Use of variables, Write executable statement, Interacting with Oracle Server, Create anonymous PL/SQL block,Sequences 2. **Control Structure in PL/SQL**- Using while loop, Do loop, For loop, Use of GOTO statement 3. **Create conditional statement using PL/SQL**- Using if statement, Using if else statement, Using elsif ladder, Using case expression. 4. **Create cursor in PL/SQL**- Implicit cursor, Explicit cursor, Parameterized cursor **5. Collection and Composite Data Types** - Working with Collections,Working with Composite Data Types		**30 Hrs**

	Module 2:	
	1. Creation of Procedures in PL/SQL **2. Functions in PL/SQL** 3. **Creation of Trigger –** Create Row level trigger, Create Statement level trigger, Create instead of trigger 4. **Handling exceptions**- Creation of user defined exception, Creation of system defined exception 5. **Creation of Package in PL/SQL**	**30 Hrs**
10	**Text Books** 1. Programming with PL/SQL for Beginners , H. Dand, R. Patil and T. Sambare, X –Team 2. Oracle pl/sql Programming ,Feuerstein, S., & Pribyl, B. ," O'Reilly Media, Inc.".	
11	**Reference Books** 1. Oracle Database PL/SQL Language Reference, 12c Release 1 (12.1) E50727-04 , Alpern, D., Belden, E., Agrawal, S., Baer, H., Castledine, S., Chang, T., & Yang, M. 2. Oracle PL/SQL for dummies , Rosenblum, M., & Dorsey, P. (2006), John Wiley & Sons. 3. PL/SQL Programming ,Ivan Bayross, BPB	
12	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**
13	**Continuous Evaluation through:** Students are expected to attend each practical and submit the written practical of the previous session. Performing Practical and writeup submission will be continuous internal evaluation. 2.5 marks can be awarded for each practical performance and writeup submission totalling to 50 marks and can be converted to 20 marks.	30 marks practical exam of 2 hours duration
14	**Format of Question Paper: Duration 2 hours. Certified copy of Journal is compulsory to appear for the practical examination** Practical Slip: Q1. From Module 1 13 marks Q2. From Module 2 12marks Q3. Journal and Viva 05 marks	

QUESTION PAPER PATTERN

(External and Internal)

<table>
<tr><td rowspan="5">I</td><td colspan="2">A Theory of 2 credits is evaluated for a total of 50 Marks</td></tr>
<tr><td>Internal Continuous Assessment:</td><td>40%[20 Marks]</td></tr>
<tr><td colspan="2">Continuous Evaluation through:
Class test of 1 of 15 marks
Class test of 2 of 15 marks
Average of the two: 15 marks

Quizzes/ Presentations/ Assignments: 5 marks
Total: 20 marks</td></tr>
<tr><td colspan="2">External Semester End Examination: 60%[30 Marks]</td></tr>
<tr><td colspan="2">Format of Question Paper: (Semester End Examination : 30 Marks. Duration:1 hour)
Q1: Attempt any two (out of four) from Module 1 (15 marks)
Q2: Attempt any two (out of four) from Module 2 (15 marks)</td></tr>
<tr><td rowspan="5">II</td><td colspan="2">A Practical of 2 credits is evaluated for a total of 50 Marks</td></tr>
<tr><td>Internal Continuous Assessment:</td><td>40%[20 Mrks]</td></tr>
<tr><td colspan="2">Continuous Evaluation through:
Students are expected to attend each practical and submit the written practical of the previous session. Performing Practical and writeup submission will be continuous internal evaluation. 2.5 marks can be awarded for each practical performance and writeup submission totalling to 50 marks and can be converted to 20 marks.</td></tr>
<tr><td colspan="2">Semester End Examination: 60%[30 Marks]</td></tr>
<tr><td colspan="2">Format of Question Paper: Duration 2 hours. Certified copy of Journal is compulsory to appear for the practical examination(30 Marks)
Practical Slip:
Q1. From Module 1 13 marks
Q2. From Module 2 12marks
Q3. Journal and Viva 05 marks</td></tr>
</table>

Examination and Standard of Passing:

Regulations regarding the scheme of exams, number of credits and standard of passing will be as prescribed by the University of Mumbai.

A student is said to have passed if he/she secures 40% of marks allotted in each head of passing. External evaluation of 30 marks and Internal evaluation of 20 marks are treated as separate heads of passing.

The Ten Point Grading System prescribed by the University of Mumbai will be as follows:

Letter Grades and Grade Points

Semester GPA/ Program CGPA Semester/ Program	**% of Marks**	**Alpha-Sign / Letter GradeResult**	**Grade Points**
9.00-10.00	90.0-100	O (Outstanding)	10
8.00-<9.00	80.0-<90.0	A+ (Excellent)	9
7.00-<8.00	70.0-<80.0	A (Very Good)	8
6.00-<7.00	60.0-<70.0	B+ (Good)	7
5.50-<6.00	55.0-<60.0	B (Above Average)	6
5.00-<5.50	50.0-<55.0	C (Average)	5
4.00-<5.00	40.0-<50.0	P (Pass)	4
Below 4.00	Below 40	F (Fail)	0
Ab (Absent)	-	Absent	0

This syllabus is applicable to IDOL students as well, w.e.f. 2025-26

Appendix B

Justification for B.Sc. (Information Technology)

1.	Necessity for starting the course:	**A large amount of The demand for IT professionals is consistently high, and individuals with a B.Sc in IT can find opportunities in various sectors, including technology companies, healthcare, finance, government, and more.**
2.	Whether the UGC has recommended the course:	**Yes**
3.	Whether all the courses have commenced from the academic year 2024-2025	**To be implemented from 2024-2025 onwards**
4.	The courses started by the University are self-financed, whether adequate number of eligible permanent faculties are available?:	**Self-financed** **Yes. Some experts are called as visiting faculties**
5.	To give details regarding the duration of the Course and is it possible to compress the course?:	**4 years. Not possible to compress the program**
6.	The intake capacity of each course and no. of admissions given in the current academic year:	**60 seats for one division. Admissions will be held from 2024-2025 onwards**
7.	Opportunities of Employability / Employment available after undertaking these courses:	**B.Sc in Information Technology can open up a wide range of opportunities and employment prospects across various industries. Additionally, as technology continues to advance, new roles and specialties within the IT field are continually emerging, providing diverse career paths for IT graduates.**

Sign of Chairperson
Dr. Mrs. R. Srivaramangai
Ad-hoc BoS (IT)

Sign of the
Offg. Associate Dean
Dr. Madhav R. Rajwade
Faculty of Science & Technology

Sign of Offg. Dean,
Prof. Shivram S. Garje
Faculty of Science & Technology