Discipline:	Science 🛛 Arts, Humanities & Social Science						ience				
	Commerce		BBA			BCA	L	[√]			
Subject Name:	ВСА										
Subject Code:	UBCAMAJ	35009	(Will b	e provide	d by the	e Univ	versity)				
Semester:	Semester I Semester II Semester III Semester IV										
	Semester V $[]$ Semester VI \square Semester VII \square Semester VIII \square										
Course Name:	THEORY OF (COMPUT	ATION								
Course Code:				(Will be	provid	ed by	the Univer	sity)			
Course Credit:	Theoretical	3		Practio	cal/Tuto	orial	1				
Marks Allotted:	Theoretical	60		Practical	/Tutori	al	0				
	Continuing E	valuation	10	Attendar	nce		5				
Course Type (tick the c	orrect alternati	ves):									
Major Core		[√]		AEC							
Interdisciplinar	y/ DSE			SEC							
Minor / Generic	c Elective			VAC							
Research Project/Dissertation \Box Vocational \Box											
Is the course focused or	n employability	/ entrepr	eneurship	o?	YES [√]	NO					
Is the course focused on imparting life skill? YES \square NO $[]$											
Is the course based on Activity? YES \square NO $[]$											
Remarks by Chairman,	UG BOS, if an	У									
UG BOS Meeting Refer	rence Number [.]						Date:				

Course Code: UBCAMAJ35009

Course Name: THEORY OF COMPUTATION

Brief Course Description:

The Theory of Computation course explore into the foundational aspects of computer science, focusing on the study of formal languages, automata, and Turing machines. It explores the theoretical underpinnings of what can be computed and how efficiently computations can be performed. This course is crucial for understanding the limits of computation and the computational complexity of various problems.

Prerequisite(s) and/or Note(s):

Students interested in this course should have completed introductory courses in Discrete Mathematics and Algorithms. A solid grasp of mathematical logic and the ability to construct and understand mathematical proofs are essential for success in this course.

Course Objectives:

The primary objectives of the Theory of Computation course are to impart a deep understanding of the basic concepts of computation, to familiarize students with various computational models, and to enable them to analyze the complexity of computational problems. By the end of the course, students will be well-versed in the theoretical aspects of computer science that are fundamental to advanced study and research in the field.

Knowledge acquired:

Students will gain a comprehensive understanding of automata theory, including finite automata, context-free grammars, and pushdown automata. They will learn about the hierarchy of formal languages and their corresponding automata, as well as the theoretical framework of Turing machines, which are central to the study of computation. Additionally, students will acquire knowledge about the boundaries of what can be computed and the inherent limitations of computational processes.

Skills gained:

The course equips students with the skills to design and analyze various types of automata and grammars. They will develop problem-solving abilities related to the computational complexity of problems and gain proficiency in constructing and proving theorems. These skills are essential for tackling complex problems in computer science and for conducting theoretical research.

Competency Developed:

Upon completing the course, students will have developed the competence to identify and solve intricate computational problems. They will have honed their abstract thinking and formal reasoning abilities, enabling them to apply theoretical concepts to real-world scenarios. This competency is vital for advancing in fields such as algorithm design, software development, and computer science research.

3rd Year: Semester 5

UBCAMAJ35009: THEORY OF COMPUTATION

[Credits:3, Lecture :45]

Unit 1: Languages (8 Lectures)

Alphabets, string, language, Basic Operations on language, Concatenation, Kleene Star

Unit 2: Finite Automata and Regular Languages (15 Lectures)

Regular Expressions, Transition Graphs, Deterministic and non-deterministic finite automata, NFA to DFA Conversion, Regular languages and their relationship with finite automata, Pumping lemma and closure properties of regular languages.

Unit 3: Context free languages (12 Lectures)

Context free grammars, parse trees, ambiguities in grammars and languages, Pushdown automata (Deterministic and Non-deterministic), Pumping Lemma, Properties of context free languages, normal forms.

Unit 4: Turing Machines and Models of Computations (10 Lectures)

RAM, Turing Machine as a model of computation, Universal Turing Machine, Language, recursively enumerable and recursive languages.

Suggested Readings

- 1. Daniel I.A.Cohen, Introduction to computer theory, John Wiley, 1996
- 2. Lewis & Papadimitriou, Elements of the theory of computation, PHI 1997.
- 3. Hoperoft, Aho, Ullman, Introduction to Automata theory, Language & Computation **3rd** Edition, PearsonEducation. 2006
- 4. P. Linz, An Introduction to Formal Language and Automata 4th edition Publication Jones Bartlett, 2006

UBCAMAJ35009: Theory of Computation Tutorial [Credits:1, L

[Credits:1, Lecture Hours:15]

Theory of Computation tutorial as assigned and advised by teacher(s).

Discipline:	Science		Arts, Humanities & Social Science							
	Commerce		BBA		BCA	۱.	[√]			
Subject Name:	BCA									
Subject Code:	UBCAMAJ35010 (Will b			e provided by the University)						
Semester:	Semester I	Semest	ter II 🗌	Semester III 🗆	ester IV 🗆					
	Semester V $[]$ Semester VI \Box Semester VII \Box Semester V									
Course Name:	WEBSITE DESIGN WITH HTML AND PHP									
Course Code:		(Will be provided by the University)								
Course Credit:	Theoretical	3		Practical/Tutorial		1				
Marks Allotted:	Theoretical	40	Practical/Tutorial		ial	20				
	Continuing l	Evaluation	10	Attendance	l	5				
Course Type (tick the c	orrect alterna	tives):								
Major Core		[√]		AEC						
Interdisciplinary/ DSE				SEC						
Minor / Generic Elective				VAC						
Research Project/Dissertation				Vocational						
Is the course focused or	n employabilit	ty / entrepre	eneurshij	p? YES [$$] NO					
Is the course focused on imparting life skill? YES [$$] NO \square										
Is the course based on Activity? YES $[\sqrt{]}$ NO \square										
Remarks by Chairman,	UG BOS, if a	iny								
	X Y 1						[
UG BOS Meeting Refe	rence Number	r:				Date:				

Course Code: UBCAMAJ35010

Course Name: WEBSITE DESIGN WITH HTML AND PHP

Brief Course Description:

The Website Design with HTML and PHP course offers a comprehensive introduction to web development, focusing on the fundamental aspects of creating dynamic and interactive websites. Students will learn how to structure web content using HTML and enhance functionality with PHP, providing them with the essential skills to build and maintain modern websites. This course is designed for those interested in web development and eager to understand the synergy between front-end and back-end technologies.

Prerequisite(s) and/or Note(s):

To enroll in this course, students should have a basic understanding of computer operations and familiarity with the internet. Prior experience with any programming language is beneficial but not mandatory. This course is suitable for beginners and those looking to expand their knowledge in web development.

Course Objectives:

The primary objectives of the Website Design with HTML and PHP course are to provide students with a solid foundation in HTML and PHP, enabling them to create well-structured and interactive web pages. The course aims to teach students how to design user-friendly websites, implement server-side scripting, and manage web forms and databases. By the end of the course, students will be proficient in developing dynamic websites that meet modern web standards.

Knowledge acquired:

Students will acquire an in-depth understanding of HTML, including elements, attributes, and the structure of web pages. They will learn about the basics of PHP, including syntax, variables, control structures, and functions. The course also covers integrating HTML with PHP to create dynamic content, handling user input through forms, and interacting with databases using PHP. Additionally, students will gain knowledge about web hosting, domain names, and deploying websites.

Skills gained:

Throughout the course, students will develop practical skills in designing and coding web pages using HTML and PHP. They will learn how to create and style web forms, validate user input, and manage data flow between the client and server. Students will also gain experience in troubleshooting and debugging code, ensuring their websites function smoothly. Furthermore, they will become adept at using development tools and environments to streamline the web development process.

Competency Developed:

Upon completing the course, students will have developed the competency to design, develop, and deploy professional-quality websites. They will be capable of creating responsive and interactive web pages, implementing server-side logic, and managing data through web forms and databases. This competency will enable students to pursue careers in web development, work as freelance web designers, or enhance their skill set for personal projects and business ventures. The course equips students with the foundational skills necessary for advanced study and specialization in web technologies.

3Rd Year: Semester 5

UBCAMAJ35010: WEBSITE DESIGN WITH HTML AND PHP [Credits: 3, Lectures:45]

Unit 1: Introduction (5 Lectures)

Introduction to Markup Languages and HTML, need and use; the Head, the Body, Colors, Attributes, Lists, ordered and unordered

Unit 2: Links, Images and Tables (5 Lectures)

Introduction; Relative Links, Absolute Links; Link Attributes; Using the ID Attribute to Link Within a Document; Putting an Image on a Page, Using Images as Links, Putting an Image in the Background, Creating a Table, Table Headers, Captions, Spanning Multiple Columns, Styling Table

Unit 3: Introduction to XML (5 Lectures)

Introduction to XML and its Goals, XML Structure and Syntax, Document classes and Rules, Scripting XML, XMLas Data, Linking with XML, XSL –Style Sheet Basics, XSL basics, XSL style sheets. (3L)

Unit 4: Introduction to PHP (5 Lectures)

PHP introduction, inventions and versions, scope, important tools and software requirements (like Web Server, Database, Editors etc.), Basic Syntax, PHP variables and constants, Types of data in PHP, Expressions, scopes of a variable (local, global), PHP Operators: Arithmetic, Assignment, Relational, Logical operators, Bitwise, ternary and MOD operator, PHP operator Precedence and associatively

Unit 5: Handling HTML form with PHP (5 Lectures)

Basic Input and Attributes, Other Kinds of Inputs, Styling forms with CSS, Where to Go from Here Capturing FormData, GET and POST form methods, Dealing with multi value fields, Redirecting a form after submission

Unit 6: PHP conditional events and Loops (5 Lectures)

PHP IF Else conditional statements (Nested IF and Else), Switch case, while, For and Do While Loops, Goto,Break, Continue and exit

Unit 7: PHP Functions (5 Lectures)

Function, Need of Function, declaration and calling of a function, PHP Function with arguments, Default Argumentsin Function, Function argument with call by value, call by reference, Scope of Function Global and Local

Unit 8: String Manipulation and Regular Expression (5 Lectures)

Creating and accessing String, Searching & Replacing String; Formatting, joining and splitting String , String Related Library functions; Use and advantage of regular expression over inbuilt function; Use of preg_match(), preg_replace(), preg_split() functions in regular expression

Unit 9: Array (5 Lectures)

Anatomy of an Array, Creating index based and Associative array, Accessing array; Looping with Index based array, with associative array using each() and foreach(); Some useful Library function

Suggested Readings

- 1. Virginia DeBolt, Integrated HTML and CSS A Smarter, Faster Way to Learn, Wiley / Sybex, 2006
- 2. Cassidy Williams, Camryn Williams Introduction to HTML and CSS, O'Reilly, 2015
- 3. XML in action web technology by William J. Pardi
- 4. Step by Step XML by Michael J. Young
- 5. Steven Holzner, "PHP: The Complete Reference Paperback", McGraw Hill Education (India), 2007.
- 6. Timothy Boronczyk, Martin E. Psinas, "PHP and MYSQL (Create-Modify-Reuse)", Wiley India PrivateLimited, 2008.
- 7. Robin Nixon, "Learning PHP, MySQL, JavaScript, CSS & HTML5", 3rd Edition Paperback, O'reilly, 2014.
- Luke Welling, Laura Thompson, PHP and MySQL Web Development", 4th Edition, Addition Paperback, Addison-Wesley Professional, 2008.
 David Sklar, Adam Trachtenberg, "PHP Cookbook: Solutions & Examples for PHP Programmers", 2014.

UBCAMAJ35010: WEBSITE DESIGN WITH HTML AND PHP [Credit:1, Lab Hours:30]

Students are advised to do laboratory/practical practice not limited to, but including the following types of problems:

- 1. Create an HTML document with the different formatting options i.e. Bold, Italics, Underline, Headings (Using H1 to H6 heading styles), Font (Type, Size and Color), Background (Colored background/Image in background), Paragraph, Line Break, Horizontal Rule, Pre tag.
- 2. Create an HTML document which consists of: Ordered List, Unordered List, Nested ordered and/or unorderedList and Image
- 3. Create an HTML document which implements Internal linking as well as External linking.
- 4. Create a table using HTML demonstrating use of columns and rows, merging multiple rows and/or columnsetc.with data and image values and contents with hyperlinking
- Create a typical student data capture form for the purpose of admission to your college using different types of HTML controls i.e. Text Box, Option/radio buttons, Check boxes, Reset and Submit buttons etc.
- 6. Create HTML documents having multiple frames in different possible formats/organization
- 7. Information Structure: In this exercise, student will practice identifying the structure of an information object
- 8. Deconstructing an XML Document: In this exercise, student will practice identifying the explicit structure within an XML document. In a sense, this is the reverse of what you did in Exercise #7.
- 9. Creating XML Markup: In this exercise, create some XML markup based on the tree representation fromprevious exercise and the content from the original sample document.
- 10. Create a PHP page using functions for comparing three integers and print the largest number.
- 11. Write a function to calculate the factorial of a number (non-negative integer). The function accept the numberas an argument.
- 12. WAP to check whether the given number is prime or not.
- 13. Create a PHP page which accepts string from user. After submission that page displays the reverse ofprovided string.
- 14. Write a PHP function that checks if a string is all lower case
- 15. Write a PHP script that checks whether a passed string is palindrome or not? (A palindrome is word, phrase, or sequence that reads the same backward as forward, e.g., madam or nurses run)
- 16. WAP to sort an array.
- 17. Write a PHP script that removes the whitespaces from astring. Sample string: 'The quick " " brown fox' Expected Output : The quick brown fox
- 18. Write a PHP script that finds out the sum of first n odd numbers.
- 19. Create a login page having user name and password. On clicking submit, a welcome message should be displayed if the user is already registered (i.e.name is present in the database) otherwise error message should be displayed.
- 20. Write a PHP script that checks if a string contains another string.
- 21. Create a simple 'birthday countdown' script, the script will count the number of days between current day andbirth day.
- 22. Create a script to construct the following pattern, using nested for loop.

. ** ***

Discipline:	Science		Arts, H	umanities &	k Social So	cience 🗆				
	Commerce		BBA		BCA	• [√]				
Subject Name:	ВСА									
Subject Code:	UBCAMAJ350	11	(Will be	e provided b	by the Univ	versity)				
Semester:	Semester I 🗆	Semest	ter II 🗆	Semester I	II 🗆 Sem	ester IV 🗆				
	Semester V $[]$ Semester VI \square Semester VII \square Semester VIII \square									
Course Name:	MICROPROCE	MICROPROCESSOR FUNDAMENTALS AND PROGRAMMING								
Course Code:		(Will be provided by the								
University) Course Credit: Theoretical 1										
Marks Allotted:	Theoretical	40		Practical/T	utorial	20				
	Continuing Ev	aluation	10	Attendance	e	5				
Course Type (tick the c	orrect alternativ	es):								
Major Core				AEC						
Interdisciplinar	y/ DSE			SEC	[√]					
Minor / Generic	c Elective			VAC						
Research Project/Dissertation				Vocational						
Is the course focused on employability / entrepreneurship? YES $[\sqrt{]}$ NO \square										
Is the course focused or	YES	S [√] NO								
Is the course based on Activity?				YES	S [√] NO					
Remarks by Chairman, UG BOS, if any										

UG BOS Meeting Reference Number:

Date: Prepared by CIRM

Course Code: UBCAMAJ35011 Course Name: MICROPROCESSOR FUNDAMENTALS AND PROGRAMMING

Brief Course Description:

The Microprocessor Fundamentals and Programming course provides a thorough introduction to the architecture, functioning, and programming of microprocessors. Students will explore the internal workings of microprocessors, learning how they interact with memory and peripherals to perform various tasks. This course emphasizes both theoretical concepts and practical programming skills, enabling students to develop a solid understanding of microprocessor-based systems.

Prerequisite(s) and/or Note(s):

Students should have a background in basic electronics and digital logic before enrolling in this course. Familiarity with assembly language or a high-level programming language is advantageous. This course is ideal for those pursuing studies in computer engineering, electrical engineering, or related fields.

Course Objectives:

The primary objectives of the Microprocessor Fundamentals and Programming course are to provide students with a comprehensive understanding of microprocessor architecture and operation. The course aims to teach students how to program microprocessors in assembly language, interface with various hardware components, and design simple microprocessor-based systems. By the end of the course, students will be equipped with the knowledge and skills to develop and troubleshoot microprocessor applications.

Knowledge acquired:

Students will gain detailed knowledge of microprocessor architecture, including the CPU, memory hierarchy, and input/output mechanisms. They will learn about instruction sets, addressing modes, and the execution of instructions. The course covers the principles of assembly language programming, allowing students to write and understand low-level code. Additionally, students will learn about interfacing microprocessors with external devices, such as sensors and actuators, and will gain an understanding of interrupts and direct memory access (DMA).

Skills gained:

Throughout the course, students will develop practical skills in programming microprocessors using assembly language. They will learn how to write efficient code to control hardware, manage data, and perform arithmetic and logic operations. Students will also gain experience in debugging and optimizing assembly programs. Furthermore, they will acquire skills in designing and implementing interfaces between the microprocessor and peripheral devices, ensuring effective communication and control.

Competency Developed:

Upon completing the course, students will have developed the competency to design, program, and troubleshoot microprocessor-based systems. They will be capable of developing assembly language programs to solve complex problems and interface with various hardware components. This competency will enable students to work in fields such as embedded systems, hardware design, and computer engineering. The course provides a strong foundation for advanced studies in microprocessor technology and related areas, preparing students for careers in the rapidly evolving field of computing and electronics.

3rd Year: Semester 5

UBCAMAJ35011: MICROPROCESSOR FUNDAMENTALS AND PROGRAMMING [Credits: 3, Lectures:45]

Unit I: Microprocessor 8085 Architecture (15 Lectures)

Introduction to Microprocessor – Introduction to Microprocessors, Components of Microprocessor: Registers, ALU, timing and control, CPU, I/O devices, clock, memory, bussed architecture, tri-state logic, Bus Structure, address bus, data bus and control bus. Block diagram of 8085, pin configuration of 8085, Architecture of Microprocessor 8085, Internal registers (8-bit & 16-bit), CPU, ALU, multiplexing and demultiplexing address/data bus, Instruction Register and Decoder, Timing and Control Unit, Interrupts and Serial I/O.

Unit II: Instruction Set-I (10 Lectures)

Machine Language and Assembly Language, Addressing modes, types of instruction format, Data Transfer type instructions, Arithmetic and logical instructions, Branching instructions -looping, Timing diagram for opcode fetch, memory read, memory write, I/O read, I/O write.

Unit III: Instruction Set-II and Programming (10 Lectures)

Special Instructions: Rotate instructions - stack and subroutine related instructions. Assembly Language Programs – Addition, Subtraction, Multiplication (8-bit), Division (8-bit), sorting- Ascending / Descending Order, Largest/Smallest (single byte), Addition of N numbers (single byte).

Unit IV: Memory/Io Interface and Interrupts (10 Lectures)

Memory Interface (Basics) – memory mapped I/O & I/O mapped I/O. Interrupts in 8085- Types, Generation of RST codes-Hardware, software interrupts and their function, Edge triggered and level triggered interrupts, Interrupt priority, Vectored and non-vectored interrupt -SIM and RIM instructions, Comparison of Microprocessor and Microcontroller.

Suggested Readings:

1. Microprocessor Architecture, Programming and Application with the8085, Ramesh S. Gaonakar, Penram International Publishing, Mumbai, (2011).

international Fuolishing, Multioal, (2011).

2. Fundamental of Microprocessor 8085: Architecture Programming, and Interfacing, V. Vijayendran, Viswanathan,

S., Printers & Publishers Pvt. Ltd (2009).

- 3. The 8051 Microcontroller, Architecture, Program and application, Kenneth J Ayala, Pen ram
- 4. Ram, Fundamentals of microprocessors and microcomputers Dhanpat Rai Publications, New Delhi
- 5. The 8080/85 Family: Design, Programming & Interfacing, John Uffenbeck, , PHI India.
- 6. A. K. Ray & K. M.Bhurchandani, Advance Microprocessor and Peripherals, 2ndEdition, Tata McGraw Hill, 2006
- 7. Mathur A.P., Introduction to Microprocessors. 3rd edn, Tata McGraw, New Delhi,
- 8. Muhammed Ali Mazidi, Janice Gillispie Mazidi The 8051 Microcontroller and Embedded systems
- 9. Microprocessors & Microcontrollers by B. P. Singh, Galgotia publications Pvt. Ltd.

UBCAMAJ35011: MICROPROCESSOR FUNDAMENTALS AND PROGRAMMING [Credit:1, Lab Hours: 30]

Students are advised to do laboratory/practical practice not limited to, but including the following types of problems: ASSEMBLY LANGUAGE PROGRAMMING in 8085

- 1. Write an assembly language program to check whether a number is even or odd.
- 2. Write an assembly language program to find the number of ones and the number of zeros in an 8-bit number.
- 3. Write an assembly language program to find the smaller of two numbers.
- 4. Write an assembly language program to multiply two 8-bit numbers.
- 5. Write an assembly language program to find the smallest among 10 integers stored in memory locations starting from 2050H
- 6. Write an assembly language program to find the largest among 10 integers stored in memory locations starting from 2050H
- 7. Write an assembly language program to sort 10 numbers using bubble sort.
- 8. Write an assembly language program to generate the first 10 fibonacci series and store the result at memory location starting from FC50H
- 9. Write an assembly language program to find the sum of the series $12 + 22 + 32 + 42 + \dots + 102$
- 10. Write an assembly language program to find the perfect square of any number and if the number is not a perfect square, display FFH
- 11. Write an assembly language program for linear search.
- 12. Write an assembly language program to calculate the following expression using a single register: Y = X2 + 2X + 3XZ
- 13. Write an assembly language program to find the sum of the first 10 even natural numbers.
- 14. Write an assembly language program to find the sum of the first n odd natural numbers.
- 15. Write an assembly language program to create an odd parity generator.
- 16. Write an assembly language program to create an even parity generator.
- 17. Write an assembly language program to find the sum of five 8-bit numbers.
- 18. Write an assembly language program to convert decimal to binary.
- 19. Write an assembly language program to convert octal to binary.
- 20. Write an assembly language program to convert hexadecimal to binary.
- 21. Write an assembly language program to convert hexadecimal to decimal.
- 22. Write an assembly language program to check whether an 8-bit number is palindrome or not.
- 23. Write an assembly language program to display the truth table for and AND gate.
- 24. Write an assembly language program to display the truth table for and OR gate.
- 25. Write an assembly language program to display the truth table for and XOR gate.
- 26. Write an assembly language program to perform n byte addition of two numbers.
- 27. Write an assembly language program to implement a simple sub routine call.
- 28. Write an assembly language program to check whether a number is prime or not

Discipline:	Science	Science 🛛 Arts, Humanities & Social Science							
	Commerce		BBA		BCA	A	[√]		
Subject Name:	BCA								
Subject Code:	UBCAMAJ35	012	(Will be	e provided by th	ie Univ	versity)			
Semester:	Semester I Semester II Semester IV								
	Semester V $[]$ Semester VI \square Semester VII \square Semester VIII \square								
Course Name:	PYTHON PROGRAMMING								
Course Code:	(Will be provided by the University)								
Course Credit:	Theoretical	3		Practical/Tuto	rial	1			
Marks Allotted:	Theoretical	40		Practical/Tutor	ial [20			
	Continuing E	valuatio	^{on} 10	Attendance		5			
Course Type (tick the c	orrect alternati	ves):							
Major Core		[√]		AEC					
Interdisciplinar	y/ DSE			SEC					
Minor / Generic	c Elective			VAC					
Research Project/Dissertation Vocational									
Is the course focused on employability / entrepreneurship? YES $[]$ NO \square									
Is the course focused or	YES [√] NO							
Is the course based on Activity?				YES $[]$ NO \square					
Remarks by Chairman, UG BOS, if any									

UG BOS Meeting Reference Number:

Date:

Course Code: UBCAMAJ35012 Course Name: PYTHON PROGRAMMING

Brief Course Description

The Python Programming course offers a comprehensive introduction to one of the most versatile and widelyused programming languages in the industry. Python's simplicity and readability make it an ideal language for beginners, while its powerful libraries and frameworks make it invaluable for experienced developers. This course covers fundamental programming concepts, Python syntax, and advanced topics, preparing students to use Python for various applications, from web development to data analysis.

Prerequisite(s) and/or Note(s):

No prior programming experience is required to enroll in this course, making it suitable for beginners. However, a basic understanding of computer operations and familiarity with high school-level mathematics will be beneficial. This course is ideal for students from diverse backgrounds, including those in computer science, engineering, data science, and other fields that benefit from programming skills.

Course Objectives:

The primary objectives of the Python Programming course are to provide students with a solid understanding of Python's syntax and core programming concepts. The course aims to teach students how to write clean, efficient, and well-documented Python code. Additionally, students will learn to use Python libraries and frameworks to solve real-world problems, develop web applications, and perform data analysis. By the end of the course, students will be equipped to tackle a wide range of programming challenges using Python.

Knowledge Acquired:

Students will acquire comprehensive knowledge of Python programming, starting with basic concepts such as variables, data types, and control structures. They will learn about functions, modules, and file handling, progressing to more advanced topics like object-oriented programming, exception handling, and working with libraries such as NumPy, pandas, and Matplotlib. The course also covers web development frameworks like Flask and Django, providing a broad understanding of Python's capabilities and applications.

Skills Gained:

Throughout the course, students will develop practical skills in writing Python code to perform a variety of tasks. They will learn to implement algorithms, manipulate data structures, and create user-defined functions and classes. Students will also gain experience in using Python for data analysis, visualizing data, and building web applications. Additionally, they will develop debugging and testing skills, ensuring their code is robust and reliable.

Competency Developed:

Upon completing the course, students will have developed the competency to design, develop, and maintain Python-based applications. They will be capable of using Python to solve complex problems, analyze data, and create dynamic web applications. This competency will enable students to pursue careers in software development, data science, web development, and various other fields that require strong programming skills. The course provides a strong foundation for advanced study and specialization in Python and related technologies, preparing students for success in the rapidly evolving tech industry.

3rd Year: Semester 5

UBCAMAJ35012: PYTHON PROGRAMMING

[Credits: 3, Lectures:45]

Unit 1: Planning the Computer Program (8 Lectures)

Concept of problem solving, Problem definition, Program design, Debugging, Types of errors in programming, Documentation.

Unit 2: Techniques of Problem Solving (8 Lectures)

Flowcharting, decision table, algorithms, Structured programming concepts, Programming methodologies viz. top-down and bottom-up programming.

Unit 3: Overview of Programming (5 Lectures)

Structure of a Python Program, Elements of Python.

Unit 4: Introduction to Python (12 Lectures)

Python Interpreter, Using Python as calculator, Python shell, Indentation. Atoms, Identifiers and keywords, Literals, Strings, Operators (Arithmetic operator, Relational operator, Logical or Boolean operator, Assignment, Operator, Ternary operator, Bit wise operator, Increment or Decrement operator)

Unit 5: Creating Python Programs (12 Lectures)

Input and Output Statements, Control statements (Branching, Looping, Conditional Statement, Exit function, Difference between break, continue and pass.), Defining Functions, default arguments.

Suggested Readings

1. Yashavant Kanetkar and Aditya Kanetkar, Let Us Python, BPB, 3rd Edition.

2. Sheetal Taneja and Naveen Kumar, Python Programming- A modular approach with Graphics, Database, Mobile and Web applications, Pearson, Sixteenth Impression-2023,

3. T. Budd, Exploring Python, TMH, 1st Ed, 2011

4. Python Tutorial/Documentation www.python.or 2015

5. Allen Downey, Jeffrey Elkner, Chris Meyers, How to think like a computer scientist: learning with Python , Freely available online.2012

UBCAMAJ35012: PYTHON PROGRAMMING

[Credit:1, Lab Hours: 30]

Students are advised to do laboratory/practical practice not limited to, but including the following types of problems:

1. Write a menu driven program to convert the given temperature from Fahrenheit to Celsius and vice versa depending upon users' choice.

2. WAP to calculate total marks, percentage and grade of a student. Marks obtained in each of the three subjects are to be input by the user. Assign grades according to the following criteria :

- a. Grade A: Percentage >=80
- b. Grade B: Percentage>=70 and <80
- c. Grade C: Percentage>=60 and <70
- d. Grade D: Percentage>=40 and <60
- e. Grade E: Percentage<40

3. Write a menu-driven program, using user-defined functions to find the area of rectangle, square, circle and triangle by accepting suitable input parameters from user.

4. WAP to display the first n terms of Fibonacci series.

- 5. WAP to find factorial of the given number.
- 6. WAP to implement the use of arrays in Python.
- 7. WAP to implement String Manipulation in python in Python.
- 8. WAP to find sum of the following series for n terms: $1 2/2! + 3/3! \cdots n/n!$
- 9. WAP to calculate the sum and product of two compatible matrices.
- 10. WAP to create Class and Objects in Python.
- 12. WAP to implement Data Hiding in Python.
- 13. WAP to implement constructor and destructor for a class in Python.
- 14. WAP to implement constructor and destructor in Python.
- 15. WAP to implement different types of inheritance in Python.
- 16. WAP to implement concept of Overriding in Python.
- 17. Write programs to create mathematical 3D objects using class. a. curve b. sphere c. cone d. arrow e. ring f. cylinder
- 18. WAP to Check if a Number is Positive, Negative or 0
- 19. WAP to Check leap year or not.
- 20. WAP to display calendar of the given month and year.