Python Programming

Course Code: DI01016011

Course Information

Field	Details
Program	Diploma in Engineering
Branch	Information Technology
Level	Diploma
Semester	1
Academic Year	2024-2025
Category	PCC
Prerequisites	Basic computer skills, including the ability to write basic statements and expressions.

Rationale

Computer programming skills are now becoming part of basic education as these skills are increasing of vital importance for future job and career prospects. The Python programming language is one of the most popular programming languages worldwide. The course emphasizes the use of python programming in multiple domains. Python is a modern language for writing compact codes specifically for programming Server-side web apps, Data Analytics and Machine Learning, an important Artificial Intelligence domain. Furthermore, Python has gained popularity in scientific computing, production tools and game programming. This course focuses on developing python programming to do a variety of programming tasks where the students are encouraged to develop basic applications using different open source tools. At the end of the course, the student will be developing adequate basic programming skills using python language.

Course Outcomes

After completion of the course, students will be able to:

No.	Course Outcomes	RBT Level
CO1	Prepare flowchart and algorithm for solving computing problems.	Apply
CO2	Develop python programs to solve simple problems.	Apply
CO3	Apply control structure feature of python for developing programs.	Apply
CO4	Develop programs in Python using built-in functions, modules, and library functions.	Apply
CO5	Develop python programs applying strings and lists manipulation concepts.	Apply

Teaching and Examination Scheme

Teaching Scheme (Hours)				Assessment Pattern (Marks)				
L	T	PR	C	Theory ESE	Theory CA	Practical CA	Practical ESE	Total
3	0	2	4	70	30	20	30	150

Course Content

Unit No.	Content	Hours	Weightage (%)
1. Problem Solving using Flowchart and Algorithm	1.1 Introduction, Steps for problem-solving, Algorithm and its characteristics, Importance of algorithm.		
	1.2 Symbolic representation of a flowchart, Importance and Limitations of flowchart, Flow of control		
	1.3 Problem solving using pseudocode	5	11	
2. Basics of Python	2.1 Introduction to python, Python features, Applications of python programming		
	2.2 Python installation 2.3 Basic structure of python program, Python Comments, Keywords, identifiers, variables, Data types, and Operators.		
	2.4 Type Conversion	10	17	
3. Flow of Control	3.1 Introduction to Flow of Control		
	3.2 Selection • If statement • Elif statement • Nested if statement • match statement 3.3 Repetition • For loop • While loop • Nested loop		
	3.4 break, continue, and pass Statements	10	24	
4. Functions	4.1 Introduction to Functions		
	4.2 User Defined Functions • Arguments and Parameters 4.3 Scope of a Variable • Global Variable • Local Variable Python Standard Library • Built-in functions • Input or output - input(), print()		

Unit No.	Content	Hours	Weightage (%)
• Mathematical Functions - abs(), divmod(), max(), min(), pow(), sum() • Module math random statistics	10	24	
5. Strings and Lists 5.2 Strings Methods and Built-in Functions 5.3 Introduction to List and its Operations 5.4 List Methods and Built-in Functions • Nested and Copying Lists 5.5 List as Arguments to Function	5.1 Introduction to Strings, String Operations, Traversing a String 10	24	

Suggested Course Practical List

Sr. No	Practical Outcomes (PrOs)	Unit No.	Hrs.
1 • Find the sum of two given numbers. • Find a maximum out of two given numbers.	Prepare flowchart and algorithm for a given problem.		

Sr. No	Practical Outcomes (PrOs)	Unit No.	Hrs.
• Find whether a given number is odd or even. • Find a maximum out of three given numbers.	1	2	
2 2	Install \	configure python software and Create a program to print your name, date of birth and mobile number.	2
3	Develop a program to identify data-types in python.	2	2
4	Create programs for mathematical operations and conversions.		

Sr. No	Practical Outcomes (PrOs)	Unit No.	Hrs.
• 1) Create a program to read three numbers from the user and find the average of the numbers.			
• 2) Create a program to convert temperature from Fahrenheit to Celsius unit using eq: C=(F-32)/1.8	2	2	
5	Create programs for conditional statements and comparison operations.		

Sr. No	Practical Outcomes (PrOs)	Unit No.	Hrs.
• 1) Create a program to identify whether the scanned number is even or odd and print an appropriate message.			
• 2) Create a program to find a maximum number among the given three numbers.	3	2	
6	Develop a program to show whether the entered number is prime or not.	3	2
7	Develop a program to print odd and even numbers from 1 to N numbers. (Where N is an integer number entered by the user)	3	2
8	Develop a program to demonstrate the use of break, continue and pass statements.	3	2
9	Develop user-defined functions for mathematical operations.		

Sr. No	Practical Outcomes (PrOs)	Unit No.	Hrs.
• 1) Develop a user-defined function to find the factorial of a given number.			
• 2) Create a user-defined function to print the Fibonacci series of 0 to N numbers. (Where N is an integer number and passed as an argument)	4	2	
10	Write a program using the function that reverses the entered value.	4	2
11	Write a program that determines whether a given number is an Armstrong number or not using a user-defined function.	4	2
12	Write programs for string manipulation operations.		

Sr. No	Practical Outcomes (PrOs)	Unit No.	Hrs.
• 1) Write a program to reverse words in a given sentence. • 2) Write a program to check if a sub-string is present in a given string.			

Sr. No	Practical Outcomes (PrOs)	Unit No.	Hrs.
• 3) Write a program to count and display the number of vowels, consonants, uppercase, lowercase characters in a string.	5	2	
13 • 1) Create a program to find the sum of all elements in a list using a loop.	Create programs for list operations and analysis.		

Sr. No	Practical Outcomes (PrOs)	Unit No.	Hrs.
• 2) Create a program to find the smallest and largest element in a given list.	5	3	
14	Given a list saved in variable: a = [1, 8, 7, 15, 25, 36, 48, 64, 81, 95]. Write a Python program that takes this list and makes a new list that has only the even elements of this list in it.	5	3