Government College of Engineering Kannur

Name : Sanjeeb J

Roll No. : 60

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Series Exam 1(November-2024)

Ist Semester

UCEST105 - Algorithmic Thinking with Python

CS 2K24

Total Mark: 30

Total Time: 1 Hrs : 30 Mins

	Course Outcome (CO)
CO1	Explain various problem-solving strategies and describe the process of developing computational models.
CO2	Utilize Python programming concepts and operators to solve problems involving data types, mathematical operations, and input/output handling.
CO3	Construct algorithms and pseudocode to solve problems, and create flowcharts using appropriate symbols to visually represent program logic.
CO4	Implement selection and iteration structures, sequence data types, arrays, and modular programming techniques in Python to solve complex problems through decomposition and the use of functions.

PART A

Answer all Questions - Each question carries 3 marks

No.	Questions	Marks	CO
1.	Identify the most suitable problem-solving strategies for the following use cases. Justify your answer: a) You encounter a new software bug that you have not seen before, and there are no clear error messages. b) You are tasked with organizing a large conference that involves multiple tasks such as booking the venue, arranging transportation, and coordinating speakers.	3	CO1
2.	What will be the result of the following expressions in Python? a) 5 + 3 * 2 b) 10 / 2 + 3 * (2 ** 3) - 5 % 3 c) (5 > 3) and (8 < 6) or (7 == 7)	3	CO2
3.	What is pseudocode? In what scenarios would using pseudocode be more beneficial than writing actual code?	3	CO3
4.	Write a Python program to check if a number is divisible by 5.	3	CO4

PART B

No.	Questions	Marks	CO
5. a)	Briefly explain the six steps of problem-solving	4	CO1

Government College of Engineering Kannur

Name : Sanjeeb J

Roll No. : 60

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Series Exam 2 (December-2024)

Ist Semester

UCEST105 - Algorithmic Thinking with Python

CS 2K24

Total Mark: 30 **Total Time: 1 Hrs : 30 Mins**

Course Outcome (CO)	
CO4	Implement selection and iteration structures, sequence data types, arrays, and modular programming techniques in Python to solve complex problems through decomposition and the use of functions.
CO5	Utilize recursion techniques to solve problems.

PART A
Answer all questions from PART A

No.	Questions	Marks	CO
1.	Consider the following Python dictionary definition and execution. What will be the output of following code. Justify your answer. d = {1: "One", "1": "String One", True: "Boolean One"} print(d, len(d))	3	CO4
2.	Given the string: s = "MALAYALAM" Write the output of the following slicing operations: i) s[-3:] ii) s[::-1] iii) s[1:8:2]	3	CO4
3.	What is modularization in software development? Discuss its benefits.	3	CO4

4.	Identify and rectify the problem with the following recursive definition to find the greatest common divisor (GCD) of two positive integers:
	def ABC (n , m):
	if n == 2:
	return m
	else:
	return ABC(m , n mod m)
		3	CO5

PART B

No.	Questions	Marks	CO
	Answer Either Question 5 or Question 6		
5. a)	Write a Python program to input two lists from the user. Merge these lists into a third list such that: i) All even numbers appear first in the merged list, followed by all odd numbers. ii) Both even and odd numbers in the merged list should be sorted in ascending order.	4	CO4

5. b)	Write a Python program to calculate the electricity bill for a house based on the number of appliances used. Each house has the following appliances: fans, lights, washing machines, and computers. The energy consumption for each appliance is as follows: • Each fan consumes 1 unit of electricity per day. • Each light consumes 0.5 units of electricity per day. • The washing machine consumes 2 units of electricity per day. • Each computer consumes 3 units of electricity per day. The cost of 1 unit of electricity is 50 rupees. Your program should take the number of fans, lights, washing machines, and computers as input for a particular house and calculate the total electricity bill for 2 months (assuming each month has 30 days).	5	CO2
	OR		
6. a)	Identify two practical use cases for each of the following problem-solving strategies: Heuristics and Backtracking.	4	CO1
6. b)	Write a Python program to convert the time input in minutes to hours and minutes. For example, 85 minutes is 1 hour 25 minutes.	5	CO2
7. a)	Write a pseudo code/algorithm to find the factorial of a given number	4	CO3
7. b)	Write a menu-driven Python program to create a simple desktop calculator that performs basic arithmetic operations.	5	CO4
	OR		
8. a)	You visit a shop to buy a new mobile. In connection with the festive season, the shop offers a 10% discount on all mobiles. In addition, the shop also gives a flat exchange price of 1000 for old mobiles. Draw a flowchart to input the original price of the mobile and print its selling price. Note that all customers may not have an old mobile for exchange.	4	CO3
8. b)	Write a Python program to determine the grade based on the KTU grading scale using the given mark.	5	CO4

5. b)	Write a Python program that accepts the lengths of three sides of a triangle as inputs. The program should determine whether the triangle is a right triangle using the Pythagorean Theorem. Recall that in a right triangle, the square of the largest side equals the sum of the squares of the other two sides. Implement the solution using functions.	5	CO4
	OR		
6. a)	Write a menu-driven Python program to input two matrices and do the following using Numpy. (i) Find the sum of the two matrices (ii) Find the difference of the two matrices (iii) Find the product of the two matrices	4	CO4
6. b)	Write a Python program to check whether a given number is a valid mobile number or not. Implement the solution using functions. The program should validate the number based on the following rules: i) The number must contain exactly 10 digits. ii) The first digit of the number must be 7, 8, or 9.	5	CO4
	Answer Either Question 7 or Question 8		
7. a)	What are the primary differences between lists, tuples, and strings in Python in terms of mutability, usage, and methods?	3	CO4
7. b)	What is a call stack in recursion? Explain how the call stack is used to keep track of function calls in recursion with an example.	3	CO5
7. c)	Write a recursive function in Python to solve Tower of Hanoi problem.	3	CO5
	OR		
8. a)	What are keyword arguments in Python? How do they differ from positional arguments?	3	CO4
8. b)	What is circularity in recursion, and what are the methods to avoid it? Why is it important to address circularity?	3	CO5
8. c)	Write a recursive function in Python to calculate the n th Fibonacci number.	3	CO5