Generated Question Paper

Question 1

• Question 1 of 5

• Section A: Programming Fundamentals (20 marks)

• Question 1: Programming in C (12 marks)
Write a C program that takes an integer input from the user and calculates the sum of all even numbers up to that input. The program should also count the number of even numbers and display the sum and count at the end.

• Sub-Parts:

• (a) Define the necessary variables and data types for the program. (2 marks)

• (b) Write the C code to take the integer input from the user and calculate the sum of all even numbers up to that input. (6 marks)

• (c) Explain how the program counts the number of even numbers. (2 marks)

• (d) Provide a sample output of the program for an input of 10. (2 marks)

• Instructions:

• Write your answer in the space provided below.

• Use proper indentation and syntax in your C code.

• Make sure to explain your answer clearly and concisely.

• Assume that the input will always be a positive integer.

• Mark Allocation:

• (a) 2 marks

• (b) 6 marks

• (c) 2 marks

• (d) 2 marks

• Total Marks: 12 marks

• Space for Answer:

Question 2

• Question 2 of 5

• Marks: 12

• Time: 20 minutes

• Write a C program to calculate the sum of all even numbers between 1 and 50 (inclusive) using a for loop. Also, explain the concept of nested loops and provide an example of a nested loop in C.

• Part A: Program (6 marks)

Write a C program to calculate the sum of all even numbers between 1 and 50 (inclusive) using a for loop. The program should prompt the user to enter the lower and upper limits, but for the purpose of this question, assume the lower limit is 1 and the upper limit is 50.

• Part B: Nested Loops (4 marks)
Explain the concept of nested loops in C programming. Provide an example of a nested loop that prints the following pattern:
```
1 2 3
4 5 6
7 8 9
```

• Part C: Description (2 marks)
Describe the difference between a for loop and a while loop in C programming. Provide an example of when you would use each type of loop.

• Note:

• The program in Part A should be well-structured and follow good coding practices.

• The explanation of nested loops in Part B should be clear and concise.

• The example of a nested loop in Part B should be correct and demonstrate a good understanding of the concept.

• The description of for and while loops in Part C should be accurate and provide relevant examples.

Question 3

• Question 3 of 5

• Marks: 10

• Instructions:** Answer all sub-questions.

• Using Functions in C
Write a C program to calculate the area and perimeter of a rectangle using functions. The program should take the length and width of the rectangle as input from the user and then display the calculated area and perimeter.

• Sub-questions:

• a)** Define a function `calculate_area` that takes the length and width of a rectangle as parameters and returns the calculated area. (3 marks)

• b)** Define a function `calculate_perimeter` that takes the length and width of a rectangle as parameters and returns the calculated perimeter. (3 marks)

• c)** Write the main function to take the length and width of the rectangle as input from the user, call the `calculate_area` and `calculate_perimeter` functions, and display the calculated area and perimeter. (4 marks)

• Note:** Your program should be well-structured, readable, and follow good coding practices.

• Mark Allocation:

• a) `calculate_area` function: 3 marks

• b) `calculate_perimeter` function: 3 marks

• c) main function: 4 marks

• Total Marks: 10

Question 4

• Question 4 of 5

• Marks: 12

• Section A: Short Answer Questions

• Part (a) - 4 marks
Draw a flowchart to represent the logic of a program that calculates the area of a rectangle. The program should prompt the user to input the length and width of the rectangle, and then display the calculated area.

• Part (b) - 4 marks
Write a C program to implement the logic represented in the flowchart you drew in Part (a). Your program should include:

• A function to calculate the area of the rectangle

• A function to display the calculated area

• A main function to call the above functions and handle user input

• Part (c) - 4 marks
Explain the difference between a compiler and an interpreter, with an example of how each is used in programming.

• Instructions:

• Answer all parts of the question

• Use clear and concise language in your answers

• Use diagrams and flowcharts where necessary

• Ensure your program is well-structured and follows good programming practices

• Note:

• The marks allocated to each part are indicative of the relative importance of each part.

• The examiner will assess your answers based on the clarity, accuracy, and completeness of your responses.

Question 5

• Question 5 of 5

• Marks: 15

• Write a C program to calculate the sum of all even numbers in a given range.

• Part A (5 marks)
Define a function `isEven` that takes an integer as input and returns `1` if the number is even, and `0` otherwise. Use this function to calculate the sum of all even numbers in the range 1 to 20.

• Part B (5 marks)

Modify the program to take the range as input from the user. Use a `for` loop to iterate over the range and calculate the sum of all even numbers.

• Part C (5 marks)
Draw a flowchart to represent the program's logic, including the `isEven` function and the main program loop.

• Instructions:

• Write your code in C programming language.

• Use proper indentation and comments to explain your code.

• Write the flowchart in a clear and concise manner.

• Make sure to test your program with different inputs to ensure it works correctly.

• Mark Allocation:

• Part A: 5 marks

• Part B: 5 marks

• Part C: 5 marks

• Note:** You can assume that the input range will always be positive integers.