

**SOUTH EASTERN UNIVERSITY OF SRI LANKA**  
**FIRST EXAMINATION IN BACHELOR OF INFORMATION AND**  
**COMMUNICATION TECHNOLOGY - 2021/2022**  
**SEMESTER – I, APRIL / MAY – 2024.**

**SWT 11012 - FUNDAMENTALS OF PROGRAMMING**

**Answer all Questions**

**Time: 02 hours.**

**Question 01:**

Explain the fundamental concepts of structured programming and how they contribute to creating more organised and maintainable code. Provide examples using pseudocode or a simple programming language of your choice.

- a) Discuss the importance of modularity in structured programming. Explain how functions (or subroutines) enhance code organisation, reusability, and readability. Describe the concepts of function parameters and return values, and how they facilitate data flow between different parts of a program.

**(05 Marks)**

- b) Describe the three basic control structures in structured programming: sequence, selection, and iteration. Explain how each of these structures contributes to program flow and logic. Provide an example scenario where each structure would be appropriately used.

**(10 Marks)**

- c) Compare and contrast top-down and bottom-up approaches in structured program design. Explain the advantages and potential drawbacks of each approach. Provide an example of a simple programming task and outline how you would approach it using both top-down and bottom-up methodologies.

**(10 Marks)**

**[25 Marks]**

**Question 02:**

Examine the fundamental principles of algorithm design and analysis in structured programming, while exploring the crucial role of data types in representing and manipulating information efficiently.

- a) Define the term "algorithm" and explain its importance in programming. Discuss the role of pseudocode in algorithm design and provide an example of how pseudocode can be used to outline a simple sorting algorithm.

**(09 Marks)**

- b) Explore the concept of data types in programming, discussing both basic (primitive) and complex (composite) data types. Explain their roles, characteristics, and how they are used in structured programming to represent and manipulate data efficiently.

**(10 Marks)**

- c) Draw the flow chart for the following pseudocode.

The flowchart would work as follows:

1. Start the process
2. Input the marks
3. Check if marks are greater than or equal to 80
  - o If yes, output "A"
  - o If no, move to the next decision
4. Check if marks are greater than or equal to 60
  - o If yes, output "B"
  - o If no, move to the next decision
5. Check if marks are greater than or equal to 40
  - o If yes, output "C"
  - o If no, output "F"
6. End the process

**(06 Marks)**

**[25 marks]**

**Question 03:**

Explore the concept of loops and repetition in programming, discussing their importance in automating repetitive tasks and processing large amounts of data efficiently. Explain how different types of loops contribute to program flow and problem-solving.

- a) Compare and contrast the three main types of loops: for loops, while loops, and do-while loops. Explain the structure and typical use cases for each type of loop. Discuss how the choice of loop type can affect program readability and performance. Provide examples of scenarios where each type of loop would be most appropriate.

**(09 Marks)**

- b) Examine the concept of nested loops and their applications in solving complex problems. Explain how nested loops work and discuss their impact on program complexity and execution time. Provide an example of a real-world problem that can be solved efficiently using nested loops, and outline the solution using pseudocode or a programming language of your choice

**(09 Marks)**

- c) The program below contains errors. Find the errors and rewrite this program with the correct code.

```
int main {}  
(  
    int i = 0, sum = 0;  
    while (i <= 5)  
    {  
        sum +=i;  
        printf ('sum [%d] = %d\n", i, sum);  
        i++  
        return 0;  
    }
```

**(07 Marks)**

**Question 04:**

Explore the importance of file handling and standard library functions in programming, focusing on how they enhance data persistence, input/output operations, and overall program functionality.

- a) Develop a C program to solicit numerical marks from a user and subsequently compute and display the corresponding letter grade and Grade Point Average (GPA) based on the provided grading scale.

Marks Range	Grade	Grade Point
85-100	A+	4.00
70-84	A	4.00
65-69	A-	3.70
60-64	B+	3.30
55-59	B	3.00
50-54	B-	2.70
45-49	C+	2.30
40-44	C	2.00
35-39	C-	1.70
30-34	D+	1.30
25-29	D	1.00
00-24	E	0.00

(16 Marks)

- b) Enhance the previously developed C program to incorporate text file input and output functionalities. Specifically, the modified program should:
- Read a text file containing a marks table (format and content to be specified your own).
  - Process the marks data to determine corresponding letter grades and GPAs based on the provided grading scale.
  - Create a new text file named "Marksheet.txt" to store the student's name, obtained marks, calculated letter grade, and GPA.

(09 Marks)

[25 Marks]

[Total 100 marks]