The Management University of Africa

Sponsored by the Kenya Institute of Management

UNDERGRADUATE UNIVERSITY EXAMINATIONS

SCHOOL OF SCIENCE AND TECHNOLOGY

DEGREE OF BACHELOR OF BUSINESS INFORMATION TECHNOLOGY

BBIT 103 : INTRODUCTION TO COMPUTER PROGRAMMING

DATE: 8TH APRIL 2026

DURATION: 2 HOURS

MAXIMUM MARKS: 70

INSTRUCTIONS:

1. Write your registration number on the answer booklet.
2. **DO NOT** write on this question paper.
3. This paper contains **SIX (6)** questions.
4. Question **ONE** is compulsory.
5. Answer any other **THREE** questions.
6. Question **ONE** carries **25 MARKS** and the rest carry **15 MARKS** each.
7. **Write all your answers in the Examination answer booklet provided.**

QUESTION ONE

Read the Case Study below carefully and answer the questions that follow:

Case Study: Student Grade Management System

A university department wants to automate their grade management system. The system needs to process student records for a class of up to 50 students. Each student record contains:

- Student ID (integer)
- Name (string, max 50 characters)
- Marks for 5 assignments (integers, 0-100)
- Final exam score (integer, 0-100)

The system must:

1. Calculate the average assignment score for each student (assignments worth 50% of final grade)
2. Calculate final grade (50% assignments + 50% final exam)
3. Assign letter grades: A (>=80), B (>=70), C (>=60), D (>=50), F (<50)
4. Generate statistics: class average, highest score, lowest score
5. List all students who scored above the class average

Required Tasks:

a) Design and justify an appropriate structure to represent a student record. Explain your choice of data types and evaluate how this structure facilitates the required operations.

[6 marks]

b) Develop a function calculateFinalGrade() that takes a student structure as input and returns the final grade. Analyze how this function would handle edge cases (e.g., missing data).

[6 marks]

c) Create a program segment using appropriate loop constructs to read student data for N students (where N ≤ 50) and store them in an array. Evaluate which loop type is most suitable and justify your choice. **[6 marks]**

d) Design an algorithm using conditional statements to categorize students into letter grades. Compare the efficiency of using if-else versus switch statements for this task.

[7 marks]

QUESTION TWO

a) A program needs to process sensor data that arrives at different intervals. Analyze the differences between using a while loop versus a do-while loop for this scenario. Create code segments demonstrating both approaches and evaluate which is more appropriate when the sensor must be read at least once before checking continuation conditions. **[8 marks]**

b) Examine the following code segment and predict the output. Then, modify it using the break and continue statements to skip even numbers and terminate when the sum exceeds 50. Justify your modifications.

```
int sum = 0;
for(int i = 1; i <= 20; i++) {
   sum += i;
   printf("Sum: %d\n", sum);
}
```

[7 marks]

QUESTION THREE

a) Compare and contrast the use of arrays versus individual variables when storing temperature readings for 7 days of the week. Design a program that uses a 2D array to store morning and evening temperatures for each day, then calculates and displays the average temperature for each day. **[8 marks]**

b) Evaluate the advantages of passing arrays to functions by reference rather than by value in C. Create a function that accepts an integer array and its size, then sorts the array in descending order using nested loops. Explain your sorting logic.

[7 marks]

QUESTION FOUR

a) Analyze the concept of pointers in C programming. Create a program that demonstrates pointer arithmetic by:

- Declaring an integer array of 5 elements
- Using a pointer to traverse and display array elements
- Modifying array values through pointer dereferencing

Explain how pointer arithmetic differs from regular arithmetic operations.

[8 marks]

b) Compare the use of array notation versus pointer notation for accessing array elements. Write two equivalent functions: one using array subscript notation and another using pointer notation to find the maximum value in an integer array. Discuss the performance implications. **[7 marks]**

QUESTION FIVE

a) Design a comprehensive example demonstrating the use of the preprocessor in C. Your example should include:

- #define macros for constants and function-like macros
- Conditional compilation using #ifdef, #ifndef
- #include directives

Evaluate how preprocessor directives improve code maintainability and portability.

[8 marks]

b) Examine bitwise operators in C. Create a program that uses bit manipulation to:

- Check if a number is even or odd using bitwise AND
- Swap two numbers without using a temporary variable
- Set, clear, and toggle specific bits in a byte

Justify when bitwise operations are preferable to arithmetic operations. **[7 marks]**

QUESTION SIX

a) Develop a program using structures and functions to manage a library book inventory. Each book should have: title, author, ISBN, and availability status. Create functions to:

- Add a new book
- Search for a book by ISBN
- Update availability status

Analyze how structures and functions together create modular and maintainable code.

[8 marks]

b) Critique the following debugging scenario: A program compiles successfully but produces incorrect output. The program calculates compound interest but shows values much larger than expected. Design a systematic debugging strategy using:

- Printf debugging statements
- Verification of intermediate calculations
- Checking for common errors (data type issues, operator precedence, logic errors)

Demonstrate your debugging approach with a sample problematic code segment and its correction.

[7 marks]