(University of Choice)

MASINDE MULIRO UNIVERSITY OF

SCIENCE AND TECHNOLOGY

(MMUST)

MAIN CAMPUS

UNIVERSITY EXAMINATIONS

MAIN EXAM

2022/2023 ACADEMIC YEAR

FIRST YEAR SECOND SEMESTER EXAMINATION

FOR THE DEGREE OF BACHELORS OF SCIENCE IN

(COMPUTER SCIENCE)

COURSE CODE: BCS 125

COURSE TITLE: PROCEDURAL PROGRAMMING

DATE: 11/04/2023   **TIME: 12:00-2:00PM**

INSTRUCTIONS TO CANDIDATES:
Answer Questions ONE and ANY OTHER TWO.

MMUST observes ZERO tolerance to examination cheating

QUESTION ONE (COMPULSORY) [30 MARKS]

a. Why should a programmer need to understand different styles of programming promoted by different languages? **[2 marks]**

b. Using suitable examples, differentiate between declarative and imperative paradigms **[4 marks]**

c. Explain the meaning of the following: **[4 marks]**

   i   Programming domain

   ii Programming technique

   iii Programing culture

   iv Programing style

d. A leap year is divisible by 400 and not divisible by 100 or is divisible by 4. Write a program that will read year from the keyboard and check if it is a leap year or not. **[3 marks]**

The two roots of a quadratic equation $ax^2 + bx + c = 0$ can be obtained using the following formula:

$$r_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \quad \text{and} \quad r_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

Where $b^2 - 4ac$ is called the discriminant of the quadratic equation. If it is positive, the equation has two real roots. If it is zero, the equation has one root. If it is negative, the equation has no real roots. Write a program that prompts the user to enter values for a, b, and c and displays the result based on the discriminant. If the discriminant is positive, display two roots. If the discriminant is 0, display one root. Otherwise, display the equation has no real roots. **[6 marks]**

e. Consider this function declaration (prototype) and function call:

```
int f( int& x, double y );

int main()
{
int u, v;
double w = 3.3;
u = f( v, w );
return 0;
}
```

Answer these questions, assuming the code above is completed by a definition of the function

   i.   What are the formal parameters? **[1 Mark]**

   ii. What are the arguments? **[1 Mark]**

iii. Which argument(s) may be changed by the call? **[1 Mark]**

iv. Which local variable gets assigned and with what value is it assigned? **[1 Mark]**

v. Which formal parameter is also a local variable of the function f? **[1 mark]**

vi. Which parameter(s) are call-by-value and which are call-by-reference? **[2 marks]**

f. Explain using a C++ code extract explain how Jump Statements: break statement, continue statement and goto statement are used. **[6 marks]**

QUESTION TWO [20 MARKS]

a. Explain any TWO limitation of procedure oriented programming approaches that can be handled by object oriented programming approaches? **[2 marks]**

b. Explain a programming language evaluation criteria. **[2 marks]**

c. What is the trade-offs between compiled programming languages and interpreted languages **[2 marks]**

d. Explain how recursion differ from iterations **[2 marks]**

e. Using suitable C++ statements explain the THREE credentials of a recursive function? **[2 marks]**

f. Using appropriate control structures, write a C++ program to simulate the expression below for n terms of the series. **[6 marks]**

$$(1+x)^n = 1 + \frac{nx}{1!} + \frac{n(n-1)x^2}{2!} + \cdots$$

g. State the rule that determines the order in which initial values are assigned to multidimensional array elements? What value is automatically assigned to those array elements that are not explicitly initialized? **[4 marks]**

QUESTION TRHEE [20 MARKS]

a. Explain the procedure followed when using input/output streams with files, state the necessary files that are required to achieve the process. **[3 marks]**

b. Write a C++ statements that will be used to declare and open input file stream fileIn and output file stream fileOut. Attach these to files named input.dat and output.dat. **[3 marks]**

c. Assume you have opened and connected stream variables fileIn and fileOut in part (a) above. Assume further that you have finished with input file, but you want to re- read the input file. Why

is it necessary to set the stream state to reread the file from the start? Write a C++ statement for this action **[2 marks]**

d. Assume you have opened and connected stream variables fileIn and fileOut in part (a) above. Assume further that you have finished with the input and output files. Write the statements necessary to close these files. **[2 marks]**

e. Assume that your program opens a file stream, has a file connected, and writes to the file. What changes need to be made to make your program write to the screen? **[2 marks]**

f. You are writing a program. Give the necessary statements to open a file and to confirm that the file has been successfully opened for writing. Why is it important to bother to test if the file has been successfully opened? **[3 marks]**

g. You are provided with a file named data.txt having the following sequence of characters *"programming is work of logic. It is like a game, once you understand the vocabularies or the game and the rule of the game, then you are ready to go"*. Write a C++ program that will read the content of file **data.txt** character by character and print it to another file named **ouput.txt.**

[5 marks]

QUESTION FOUR [20 MARKS]

a. Simulate the output of the following code extract. **[2 marks]**

```
int  = 2;
for(;i<10; i+=3)
{
cout<<i<<" ";
}
```

b. If the three sides of a triangle are entered through the keyboard, write a program to check whether the triangle is isosceles, equilateral, scalene or right angled triangle. **[4 marks]**

c. Provide a recursive definition for the sequence 2, 4, 8, 16,. . . Write a recursive method that given n, computes the nth term of that sequence. Provide an equivalent iterative implementation and compare the two implementations. **[5 marks]**

d. Write a loop that draws a triangle of asterisks shown below **[3 marks]**

* * * * * * *

* * * * * *

* * * * *

* * * *

```
* * *
* *
*
```

e. The marks obtained by three (3) students in 5 different subjects are input through the keyboard. The student gets a division as per the following rules:

- Average percentage above or equal to 70 and less than or equal to 100 - First division
- Average percentage above or equal to 60 and less than 70 - Second Upper Division
- Average percentage above or equal to 50 and less than 60 - Second Lower Division
- Average percentage above or equal to 40 and less than 50 -Pass
- Percentage less than 40 – Fail.

Write a C++ program that calculate the division obtained by each student. The output of the program should look like the sample below (don't worry about the borders but use the correct iomanip functions) **[6 marks]**

Student	Sub1	Sub2	Sub3	Sub4	Sub5	Total	Average	Division
1	40	50	65	70	80	305	61	Second Upper Division
2	33	44	55	56	45	233	46.6	Pass
3	74	66	77	88	91	396	79.2	First division

QUESTION FIVE [20 MARKS]

a. Write a C++ code extract that demonstrates how a pointer can be passed as an argument to a function and how a pointer can be used in accessing array elements. **[4 marks]**

b. Explain how memory management is handled using appropriate C++ statements

[3 marks]

c. A year has got 12 months. The months are numbered from 1 to 12. Each month has a name with month 1 being January and month 12 being December. The names of the months are stored in a private static array months whose structure in memory is as below

months											
JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC

Write C++ code extracts or functions that:

i. Declares and initializes the array months as shown in the above structure **[2 marks]**

ii. A function that initializes a month of the year. Month is initialized to a number between 1 and 12 (1 and 12 inclusive) or initialized to 1 if the month is out of the specified range.

[2 marks]

iii. A function that returns the number of current month of the year. **[2 marks]**

iv. A function that returns the name that corresponds to the current month of the year. If current month is 1, this function returns January. **[2 marks]**

v. A function that returns the name of the next month. **[1 marks]**

vi. A function that returns the name of the previous month. **[2 marks]**

vii. A function that adds one to the current month. **[2 marks]**

viii. A function that adds number of months it receives as a parameter to the current month. [Re-use the function defined in **(vii)** above] **[2 marks]**