

70



**MASINDE MULIRO UNIVERSITY OF
SCIENCE AND TECHNOLOGY
(MMUST)**

MAIN CAMPUS

**UNIVERSITY SUPPLEMENTARY/SPECIAL EXAMINATIONS
2022/2023 ACADEMIC YEAR**

THIRD YEAR 2ND SEMESTER EXAMINATIONS

**BACHELOR OF SCIENCE IN
COMPUTER SCIENCE**

COURSE CODE: BCS 362
COURSE TITLE: GENERIC PROGRAMMING WITH C++

DATE: WEDNESDAY 19TH April 2023

TIME: 8:00 - 10:00

INSTRUCTIONS TO CANDIDATES

THIS IS AN OPEN BOOK EXAMINATION

Answer Question **ONE** and Any **OTHER 2** questions

Ensure your answers/ideas are clearly expressed

All your answers must be clearly numbered

Write in ink. Rough work can be done (in answer booklet) in pencil and will not be marked. Cross out any rough work.

Calculators, phones, tablets, computers not allowed

TIME: 2 Hours

MMUST observes ZERO tolerance to examination cheating

This Paper Consists of 06 Printed Pages, including cover page. Please Turn Over. ►

QUESTION ONE: COMPULSORY QUESTION [20 MARKS]

A playlist consist of various songs stored in different memory location. A song has three data items, its' index in the list, song title and an address to the next song. Default value of index is integer, title's default value is of type string. The last song stores the address of the first song, so that the playlist can restart when all songs are played. The number of songs in the list are not fixed and can grow or shrink on demand.

- (i) Declare a template that can be used to implement the class **PlayList** [2 Marks]
Write definition of a C++ class **PlayList** that can be used to implement the playlist. In your class
- (ii) Declare variables of a song as private members of the class. [2 Marks]
- (iii) Define an explicit default constructor that uses member initializer list to initialize song details to the value passed to it and address of the next song to null. [2 Marks]
- (iv) Define set functions **setIndex(...)** and **setTitle(...)** that can be used to initialize index and title of a song, such that their calls can be cascaded. [2 Marks]
- (v) Write the definition of functions **getIndex()** and **getTitle()** such that they return songs' index and title, respectively. [2 Marks]
- (vi) Define a function **setNext(...)** that sets the address of the next song on the playlist. [2 Marks]
- (vii) Define a function **getNext()** that returns the address of the next song, as stored by the current song. [2 Marks]
- (viii) In global namespace, write definition of function **void printList(...)** that displays the song details in the playlist. [3 Marks]
- (ix) Write a main function, in it, create a playlist of four songs with indices 1, 2, 3 and 4, then invoke the function defined in (viii) above to display the playlist.

Sample Play List

1. Zuena
2. Badder Dan
3. Yatapita by Diamond
4. Baby Calm Down

Remember, last song on the list stores the address of the first song. [3 Marks]

QUESTION TWO

[20 MARKS]

Below is register of students taking BCS 362E. The register is saved as `students.txt`

Student Register	
COM/B/01-00174/2020	Patrick Kaemba
COM/B/01-00165/2020	Kenneth Lusaku
COM/B/01-00184/2020	Hillary Barchock
COM/B/01-00171/2020	Shiella Mwanigu
COM/B/01-00167/2020	Stacey Wanjau

The student take two CATs and one end of semester examination. CAT and exam marks are saved in a file named `marks.csv` as shown below

REGNO	CAT 1	CAT 2	Exam
COM/B/01-00174/2020		23	39
COM/B/01-00165/2020	19	21	
COM/B/01-00184/2020		23	35
COM/B/01-00171/2020	24	19	35
COM/B/01-00165/2020	19	22	45

The mean of the two CATs is calculated and summed with the exam mark to find the final mark. Grades are awarded if a student has all the three marks, otherwise the grade is allocated a character **I** if a student is missing any mark. The grading system is as shown in the table below

Mark Range	Grade	Comment
70 - 100	A	Pass
60 - 69	B	Pass
50 - 59	C	Pass
40 - 49	D	Pass
0 - 39	E	Fail
Missing a mark	I	Incomplete

Write a C++ program that reads the contents of two files above and write into a file named `results.csv` such that results file looks like shown below.

REGNO	Name	CAT1	CAT2	CAT	Exam	Total	Grade	Comment
COM/B/01-00174/2020	Kaemba Patrick	-	23	-	39	-	I	Incomplete
COM/B/01-00165/2020	Lusaku Kenneth	19	21	20	-	-	I	Incomplete
COM/B/01-00184/2020	Barchock Hillary	-	23	-	35	-	I	Incomplete
COM/B/01-00171/2020	Mwanigu Shaeilla	24	19	21.5	35	56.5	C	Pass
COM/B/01-00165/2020	Wanjau Stacey	19	22	20.5	45	65.5	B	Pass

NB: The `csv` files have no spaces in-between values.

[20 Marks]

QUESTION THREE

[20 MARKS]

- (a) Two vectors are declared as shown below

```
1 vector<int, 9> data_1{4, 5, 7, 9};  
2 vector<int, 7> data_2{10, 13, 14};
```

- (i) Write C++ line of code that will insert **16** as 7^{TH} element of **data_1** without using subscript [] operator. [2 Marks]
- (ii) Write a C++ line of code that will swap the two vectors. [2 Mark]
- (iii) Using range-based for loop, write C++ lines of code to display elements of **data_2** after the line in (ii) above is executed. Write the output as will be displayed by the for loop. [4 Marks]
- (b) A positive integer is called a perfect number if it is equal to the sum of all of its positive divisors, excluding itself. For example, **6** is the first perfect number because $6 = 3 + 2 + 1$. The next is $28 = 14 + 7 + 4 + 2 + 1$. There are four perfect numbers less than **10,000**. Using a **while** loop within a **do...while** loop, write a C++ program to find and display all these four numbers and their factors. Your output should match sample output below. [4 Marks]

Question (b) Sample Output

```
1 + 2 + 3 = 6  
1 + 2 + 4 + 7 + 14 = 28
```

- (c) You are required to write a C++ program that can read two pieces of data from the user in any data type, display the inputs before and after swapping
- (i) Write a function that can swap two pieces of data of any data type it receives as parameters. [4 Marks]
- (ii) Assuming the main function is partially defined as

```
1 void* a = new string("Data");  
2 void* b = new string("Come");  
3 swap(a, b);  
4 int* x = new int(10);  
5 int* y = new int(20);  
6 swap(x, y);
```

write 4 lines of code to display values of variables **a**, **b**, **x** and **y** after calls to swap function defined in (i) above. [4 Marks]

For instance, If the user enters **0.5** as value of **x**, the main function display an output equivalent to **Sample Output 1** shown below

```
Sample Output 1
Enter value of x : 0.5
ln( 1 + 0.5 ) = 0.405465
ln( 1 - 0.5 ) = -0.693147
```

but if user enters a value greater than 1 (say user enters 1.5), output should be equivalent to **Sample Output 2** shown below

```
Sample Output 2
Enter value of x : 1.5
Input should be greater than 0 and less than 1
```

[4 Marks]

QUESTION FOUR

[20 MARKS]

Maclaurin series for natural logarithm of a number is given by

$$\ln(1 - x) = - \sum_{n=1}^{\infty} \frac{x^n}{n}$$

$$\ln(1 + x) = \sum_{n=1}^{\infty} (-1)^{n+1} \frac{x^n}{n}$$

for all $|x| < 1$. Given a partial class definition as

```
1 ...
2 class Log{
3     private:
4         ...
5     public:
6 };
```

- (i) Write Line 1 and Line 4 of the code such that class **Log** is generic and variable declared at Line 4 can take any numeric data. Variable name at Line 4 is assumed to be **x**. [2 Marks]
- (ii) Write the definition of function **void setX(...)** such that it properly initialize **x**, where $0.0 < x < 1.0$. [3 Marks]
- (iii) Write definition of get function for instance variable **x**. [2 Mark]
- (iv) Using a **while** loop, write definition of a function named **log_m_x()** such that it returns natural logarithm of **1-x** [3 Marks]
- (v) Using a **do...while** loop, write definition of a function named **log_p_x()** such that it returns natural logarithm of **1+x** [3 Marks]
- (vi) Write definition of the main function. In the function:-
 - Create an object of the class **Log**
 - Prompt user to enter value for **x** and call appropriate function of class **Log** to initialize instance variable **x**, only if $x < 1.0$ then call appropriate member functions of **Log** class, display **ln(1 + x)** and **ln(1 - x)**
 - If $x \geq 1$, display error message "**Input should be greater than 0 and less than 1**" and exit.