National University of Computer and Emerging Sciences, Lahore Campus

Course:	**OOP Lab**	**Course Code:**	**CL-217**
Program:	**BS (Computer Science)**	**Semester:**	**Fall 2019**
Duration:	**150 Minutes**	**Total Marks:**	**60(30+30)**
Paper Date:	**30 Nov 2019**	**Weight**	**40%**
Section:	**All**	**Page(s):**	**2**
Exam:	**Final Term**	**Reg. No**	

Instruction/Notes:

1. Understanding the question paper is also part of the exam, so do not ask any clarification.
2. No USB's, PHONES and INTERNET are allowed.
3. Talking/Discussion is not allowed. It is your responsibility to protect your code and save it from being copied. If you don't protect it all matching codes are considered copy/cheating cases.
4. Submission Path is:

Question # 1: Declare and implement the *abstract* class **Media**. This class will have a protected member variable **title** (of char * type) to store the title of the media item. Apart from the overloaded constructor, Media class will have a *pure virtual function* **display ()**.

Inherit three classes from the Media class, namely: **Book, Magazine**, and **CD**.

1. The Book class will have ***authorName*** *(char *)* and ***ISBN*** *(char *)* of the book.
2. The Magazine class will have ***monthName*** *(char *)* and ***year*** *(int)* of publication of the magazine.
3. The CD class will have an integer member variable to store its ***capacity*** in MBs.

Add a **Shelf** class to store a list of **Media** items. So, **Shelf** class has ***Items*** *(Media **)*, ***currSize*** *(int)*, ***maxSize*** *(int)* data members. It will have the following functions:

- `void insert (Media*);`
- `void displayContents ();`

The overloaded constructor will take an integer value as argument and initialize the **maxSize** to that value, and initialize **currSize** to 0. Constructor will also dynamically allocate an array of **Media*** through the member variable **items**.

Now, implement a main function which should ask the user how many Media items the user wants to create, and declares a **Shelf** object to store those many items. Create a menu on screen on which the user should be asked to enter 1 if he/she wants to create a Book and 2 if he/she wants to create a Magazine, 3 if he/she wants to create a CD, and 4 if he/she wants to print details of objects in the shelf.

1. If choice 1, 2, or 3 has been entered, your program should ask the user for all the attributes necessary for creating that item (Book, Magazine, or CD). Then, that item should be dynamically allocated and passed to insert method of shelf.
2. If the user has entered 4, then details of media items should be displayed by calling the **displayContents ()** function.

Question # 2: You have to design a C++ **template** function **range**, which takes a dynamic two-dimensional square matrix, its dimensions (rows, columns) size. It returns the range of values in matrix.

Range = ((max – min)/4) + min.

Note: No specialization is required for this function.

Do not take **input** from **user**. **Initialize** a 2D array in **main** and call the function and then **print** the rotated matrix **in main**.

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

Range = ((15-0)/4) + 0

Range = 3

A	B	C
H	I	D
G	F	E

Range = ((G – A) / 4) + A

Range = B

Question # 3: You have to implement amateur **Campus Management System** for **FAST-NU**.

1. Class **universityCampus** will have a ***campusname*** *(char *),* ***departments*** *(Department *).*
2. Class **Department** will have **DepartmentName (char *)**, Number of Faculty Members per department **noOfFacMembers (int)** and Number of Staff Members per department **noOfStaffMembers (int).**

Consider the driver program given below. Add all necessary **methods** to the classes, so that **driver program works properly**, without any compile, run time or logical errors. Specifically: provide the necessary **constructors** and operators. Read the comments in the driver program to get a hint that how the methods work. **Do not add useless getter /setters unless or until they are required.**

```
void main() {
    char *dpts[20] = { "CS","EE","A&F", nullptr };  // it is same to char **
    int faculty[] = { 46,33,23 }, staff[] = { 4, 3, 5 };
    UniversityCampus lhr("Lahore", dpts, faculty, staff);
    lhr.printCampusDetails();
    bool sucessFull = lhr + "Mng";      //Adds the department if not already exists
    //and sets the value of faculty and staff for this department to zero.

    lhr.addFaculty("Mng", 44);
    //if department exists number of faculty members will be added to that department.

    lhr.addStafftoDepartment("Mng", 4);
    //if department exists number of staff members will be added to that department.

    sucessFull = lhr - "A&F";
    //There must be member function in administration to find this member and delete it
    if (sucessFull)
        cout << "Department removed successfully.";
    else
        cout << "Data Not Found.\n";
}
```