National University of Computer and Emerging Sciences

Assignment

For

Object-Oriented Programming

FAST School of Computing

Submission deadline
29-Apr-2024 10:00 PM

- Make a word document with the naming convention "SECTION_ LAB#_ROLLNO" and put all your source code and snapshots of its output in it. Make sure your word file is formatted properly.
- Zip the word file with all Source code.
- Plagiarism is strictly prohibited.
- Do not discuss solutions with one another.

Task 1:

Write a program that has an abstract base class named Quad. This class should have four member data variables (floats) representing side lenghts and a pure virtual function Area. It should also have a method for setting the data variables. Derive a class Rectangle from Quad and override the Area method so that it returns the area of the Rectangle. Write a main function that creates a Rectangle and sets the side lengths. Also write a top-level function that will take a parameter of type Quad and return the value of the appropriate Area function.

Task 2:

You have been hired by a furniture store owner to develop an inventory program. He sells two main types of furniture: Beds and Chairs. Each bed or chair can be either small, medium or large. Each chair can be of different height. This height is measured in feet. There is also a special type of chair available called a Personalized Chair. Personalized chairs can have a name printed on the back. The store owner also knows C++ and has taken upon himself to construct the user interface. However, he has given you a basic driver program shown below which uses all the classes and functions that are required. He also has given you a sample output of the code. Write C++ classes in a proper hierarchy which enable the driver given below to compile and produce the given output. You cannot change the driver program at all. Your code also must not have any memory leaks.

```
#define SMALL 0
#define MEDIUM 1
#define LARGE 2
int main()
{
      const int size = 5;
      Furniture ** inventory = new Furniture * [size];
      inventory[0] = new Bed(SMALL);
      inventory[1] = new Chair(MEDIUM);
      inventory[2] = new Chair(MEDIUM,3);
      inventory[3] = new Bed(LARGE);
      inventory[4] = new PersonalizedChair(SMALL,2,"Ali");


for (int i = 0; i < size; i++)
      {
            inventory[i]->printDescription();
            delete inventory[i];
      }
      delete [] inventory;

      return 0;
```

Task 3:

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

Task 4:

You have been hired by a courier company that ships different packages all over the country. Each package must have a source location, destination location and package weight (measured in pounds). The regular shipping charges of a package is Rs. 20 per pound. Recently, the company started a new service that ships packages to the destination in less time. There are two types of packages that can be delivered using this service: 2-day package and urgent package. The 2-day package is charged a fixed additional amount other than the regular shipment charges and the urgent package is charged an additional percentage of the regular shipment charges. The company has given you a basic driver program shown below that includes all the required functionality and classes. Write C++ classes in a proper hierarchy which enable the driver given below to compile and produce the given output. You cannot change the driver program at all. Your code also must not have any memory leaks.

```
int main() {
        const int size = 5;
        package ** pkg = new package * [size];
        pkg[0] = new package("Lahore", "Karachi", 20, );
        pkg[1] = new TwoD_package("Lahore", Islamabad", 35, 200);
        pkg[2] = new Urgent_Package("Karachi", "Lahore",25, 10);
        pkg[3] = new TwoD_package("Karachi", "Islamabad", 30, 250);
        pkg[4] = new Urgent_Package("Karachi", "Peshawar", 40, 25);

        for (int i = 0; i < size; i++){
                cout<< "Package charges: "<<pkg[i]->comp_charges();
                delete pkg[i];
        }
        delete [] pkg;

        return 0;
}
```

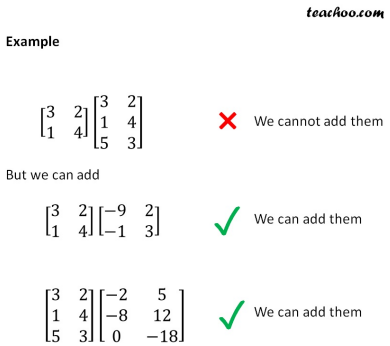
Task 5:

Your mathematician friend has asked you to create a library that can help him perform computation on Matrices. He has provided you with the following information about matrix and matrix computations.

- A matrix is a rectangular arrangement of numbers, symbols, or expressions in rows and columns. The size of a matrix is determined by the number of rows and columns it has, which is written as "# of Rows x # of columns".e.g
- You are suppose to provide follc ons for matrices

$$A = \begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix}_{2 \times 3}$$

  - Addition/Subtractoin of Two matrices i.e. A + B / A - B with



    - The two matrices can be added if the number of rows and columns of both metrics are the same i.e. 2x3 matrix can only be added to another 2x3
    - Warning message should be shown if the above rule is not followed
  - Multiplication of Matrix i.e. A * B with following rules in mind
    - The two matrices can only be multiplied if number of columns of first matrix is equal to number of rows of second column e.g be multiplied with 2x3
    - Warning message should be shown if the above rule is not followed

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \times \begin{bmatrix} j & k & l \\ m & n & o \\ p & q & r \end{bmatrix} = \begin{bmatrix} aj + bm + cp & ak + bn + cq & al + bo + cr \\ dj + em + fp & dk + en + fq & dl + eo + fr \\ gj + hm + ip & gk + hn + iq & gl + ho + ir \end{bmatrix}$$

- You also need to support scalar multiplication i.e. B * 10. / 10 * B Where all elements of B are multiplied by scalar number
- Assignment of Matrix to another matrix variable i.e. A = B
- For simplicity, I should be able to perform addition, subtraction and multiplication along with self assignment i.e. A += B, A -=B, A*=B
- You should be able to compare matrix to evaluate if they are equal i.e. A == B. The matrices are equal if
  - Matrices must have an equal number of rows and columns.
  - The corresponding elements of the matrices are the same.
- Override [] operator such that by giving A[4][5], I should be able to get value at 4th row and 5th column i.e. for the following matrix A[2][2] should return ***e***.

$$A = \begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix}_{2 \times 3}$$

- It should show warning message if any of the index is not valid i.e. for above image A[3][2] should give error because the third row does not exist

Keeping the above requirements and matrix operations in mind. You need to implement the matrix class and all the required operators over loaders so that the above operations can be performed. The matrix can have any number of rows/columns therefore, use Dynamic Memory Allocation for member variables that saves data. Then create a menu-based system in main that can be used to input matrices, perform required operations and output the results.

Task 6:

Your team has built an Agricultural land management system. This software also helps land owners to keep track of the land that they own. You are required to build class(es) that can be used to maintain the total area of lands owned by a particular farmer. You need to build classes to measure land in

- square foots
- square yards
- square metres

For all these classes you need to perform operator overloading such that following operations can be performed

- square foots +/- square foots => result in square foot
- square foots +/- square yards => converts sq. yards to sq. foot and result in sq. foot
- square foots +/- square metres => converts sq. metres to result in square foot
- square foots +/- float => converts float to sq. foot and gives result in square foot

You also need to support similar operations for square yards and square metres