CSL304- IOOM - Object Oriented Programming

Lab Assignment – 4 Batch R3 & R4

Evaluation: Wednesday September 26, 2018, at 2.00 PM

Note:

i) Input should be taken only from the user and not hard coded.

ii) You need to make all data members either private or protected.

iii) Make all methods either public or default.

Define appropriate constructors, destructors, setters and getters in particular classes.

Batch R3 & R4

1. A University and a Company have jointly taken a project. **Class University** contains name of the university, department to which the project is assigned, person to whom the project is assigned. A function display is there to display the information. **Class Company** contains name of the company, Number of Engineers assigned, amount invested to do the project. A function display is there to display the information. **Class Project** is inherited from University and Company. It contains type of project, duration of project, amount granted to complete the project. A function display displays the related information.
   Write a C++ program to implement this and display all information except amount invested by company from Project class.

2. Result of a student is dependent on his examination mark and extracurricular marks. Create four classes Student, Examination, Extracurricular, Result. The data members and methods of different classes are given below.

   Class Student:

   Data Member: Name, RollNumber;

   Method:

   get_Name();

   get_RollNo();

   display_details();

   Class Examination:

   Data Member: test1, test2;

   Method:

   Cal_average();

   Display_average();

Class Extracurricular

Data Member: craft, music

Method:

get_score();

display_total();

// To get and display the total marks in craft and music

Class Result

Data Member: total

Method:

Cal_total();

Comment();

//Calculate the total marks and display comment whether the student have passed or not.

Class Examination and Extracurricular are inherited from Student and Result is inherited from Examination and Extracurricular.

3. Imagine a publishing company that markets both book and audiocassette versions of its works. Create a class *publication* that stores the title (a string) and price (type float) of a publication. From this class derive two classes: *book*, which adds a page count (type int), and *tape*, which adds a playing time in minutes (type float). Each of these three classes should have *setter* methods to get its data from the user at the keyboard, and *getter* methods to display its data. Write a main() function to test the book and tape classes by creating instances of them, asking the user to fill in data and then displaying the data.

4. Start with the publication, book, and tape classes of Q1. Add a base class *sales* that holds an array of three floats so that it can record the Rupee sales of a particular publication for the last three months. Include a getdata()function to get three sales amounts from the user, and a getter methods to display the sales figures. Alter the book and tape classes so they are derived from both publication and sales. An object of class book or tape should input and output sales data along with its other data. Write a main() function to create a book object and a tape object and exercise their input/output capabilities.

CSL304- IOOM - Object Oriented Programming

Lab Assignment – 4 Batch R1 & R2

Evaluation: Friday September 28, 2018, at 2.00 PM

Note:

i) Input should be taken only from the user and not hard coded.

ii) You need to make all data members either private or protected.

iii) Make all methods either public or default.

Define appropriate constructors, destructors, setters and getters in particular classes.

Batch R1 & R2

1. Consider the following class definition:

```
class String
{
        private:
        enum{SZ=80;}
        char str[SZ];
        public:
        String()
        {
                str[0]='\0';
        }
        String(char s[])
        {
                strcpy(str,s);
        }
        void display() const
        {cout<<str;}
}
```

The String class in this example has a flaw: It does not protect itself if its objects are initialized to have too many characters. The SZ constant has the value 80, which can cause the str array in s to overflow, with unpredictable consequences, such as crashing the system. With String as a base class, derive a class *Pstring*(for "protected string") that prevents buffer overflow when too long string constant is used in a definition. A new constructor in the derived class should copy only SZ–1 characters into str if the string constant is longer, but copy the entire constant if it's shorter. Write a main() function to test different lengths of strings.

2. Operators in some computer languages allow you to select parts of an existing string and assign them to other strings. The Standard C++ string class offers a different approach. Using inheritance, add this capability to the Pstring class (above question). In the derived class, Pstring2, incorporate three new functions:

left(),mid(), and right()
s2.left(s1, n) // s2 is assigned the leftmost n characters from s1
s2.mid(s1, s, n) // s2 is assigned the middle n characters from s1, starting at character number s
s2.right(s1, n) // s2 is assigned the rightmost n characters from s1

You can use for loops to copy the appropriate parts of s1, character by character, to a temporary Pstring2 object, which is then returned. Have these functions return by reference, so they can be used on the left side of the equal sign to change parts of an existing string.

3. Consider the Scenario of a school data and imitate the hierarchy in a C++ program.

   Class Person
   Data Member: Name , Phone, E_mail
   Methods:
   printDetails()

   Class Teacher
   Data Member: T_id
   Subjects : Has to be a LinkedList to keep a record of all the subjects taught.

   Class NonTeachingStaff
   Data Member: T_id, Job_Role

   Class **Student**
   Data Member: Roll_no, CGPA