TUF | THE UNIVERSITY OF FAISALABAD

Object Oriented Programming – Assignment-03

Inheritance.

Instructions for Submission **20 Marks**

Submission Date: Wednesday , 22-May-2024 12:00 PM.

Submission Guidelines:

- Your code must be properly commented.
- All the steps involved in solution of each question should be written. Just Answers are not required
- This is an individual assignment. **PLAGIARISM IS NOT ACCEPTABLE!** In case of plagiarism, you will get **ZERO MARKS** for that question
- Try NOT to copy paste data from your friends, if you want to improve **Programming Skills do it by yourself**.
- Marks will be deducted if the code does not conform to the standards set out in the document **Programming Standards for Assessed Work**.
- Make a simple flowchart then break your flowchart based on input/output or condition /loops
- Your document should contain your **source code** along with **output** of your program.
- Submit your documents in **PDF Form to CR/GR before the deadline**
- These lines must be in your code before starting programming

```
// File: File Name.cpp
// Date: 02-05-2024
// Name: Your Name (Ex: Muhammad Javed)
// Registration No: Your REG No (Ex: 2018-BS-AI-072)
// Question Statement
```

Problem 1:	

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 a getdata() function to get its data from the user at the keyboard, and a putdata() function to display its data. Write a main() program to test the book and tape classes by creating instances of them, asking the user to fill in data with getdata(), and then displaying the data with putdata().

Problem 2:

Start with the publication, book, and tape classes of Question 1. Add a base class sales that holds an array of three floats so that it can record the dollar sales of a particular publication for the last three months. Include a getdata() function to get three sales amounts from the user, and a putdata() function 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.

Problem 3:

Assume that the publisher in Question 1 and 3 decides to add a third way to distribute books: on computer disk, for those who like to do their reading on their laptop. Add a disk class that, like book and tape, is derived from publication. The disk class should incorporate the same member functions as the other classes. The data item unique to this class is the disk type: either CD or DVD. You can use an enum type to store this item. The user could select the appropriate type by typing c or d.

Problem 4:

Derive a class called employee2 from the employee class in the EMPLOY program in this chapter. This new class should add a type double data item called compensation, and also an enum type called period to indicate whether the employee is paid hourly, weekly, or monthly. For simplicity you can change the manager, scientist, and laborer classes so they are derived from employee2 instead of employee. However, note that in many circumstances it might be more in the spirit of OOP to create a separate base class called compensation and three new classes manager2, scientist2, and laborer2, and use multiple inheritance to derive these three classes from the original manager, scientist, and laborer classes and from compensation. This way none of the original classes needs to be modified

Problem 5:	

Create a simple inheritance hierarchy for a Shape class, Circle class, and Rectangle class. The Shape class should be the base class, and Circle and Rectangle should be derived classes. Implement the following in C++:

Shape Class:
Attributes: color (type std::string).
Methods: A constructor to initialize the color and a method printColor to display the color.

Circle Class:
Attributes: radius (type double).
Methods: A constructor to initialize the color and radius, a method calculateArea to calculate the area of the circle (area = π * radius * radius), and a method printArea to display the area.

Rectangle Class:
Attributes: length and width (type double).

Methods: A constructor to initialize the color, length, and width, a method calculateArea to calculate the area of the rectangle (area = length * width), and a method printArea to display the area.

Problem 6:	

Design a class hierarchy for an Employee management system. The base class should be Employee with derived classes SalariedEmployee and CommissionEmployee. Each class should have appropriate data members and member functions to handle the specific attributes and behaviors of each type of employee.

Employee: Should have data members for name, employee ID, and department. It should also have member functions to get and set these values.

Salaried Employee: Inherits from Employee and adds a data member for annual Salary. It should have member functions to get and set the salary, and to calculate the monthly pay.

Commission Employee: Inherits from Employee and adds data members for sales and commission Rate. It should have member functions to get and set these values, and to calculate the total pay based on sales and commission rate.