Department of Computer Science & Engineering, NITS
CS212 - Object Oriented Programming Laboratory
Session : January - June 2024

Lab-3: Inheritance

Programming Assignments :

1. *(**Single**)*: *When only one base class is used for the derivation of a derived class. Further, derived class is not used as a base class, such a type of inheritance that has one base and derived class is known as single inheritance.*
   Design a base class "Vehicle" with member variables like "model" and "fuelType", along with member functions for basic functionalities. Derive a class "Car" with additional member variables like "numDoors" and "isConvertible", introducing specialized member functions for car-specific operations. Further, create a class "ElectricCar" inheriting from "Car" with a unique member variable "batteryCapacity" and methods related to electric vehicles. Write a program to exercise the above scenario.

2. *(**Multiple**)*: *When two or more base classes are used for the derivation of a class, this is called multiple inheritance.*
   Define a class **personType**; The common attributes of a person are the person's first and last name. The typical operations on a person's name are to set the name (separate method for first name and last name) and print the name.
   To define the class **dateType**, we need three member variablesto store the month, day number, and year. Some of the operations that need to be performed on a date are setting the date and printing the date.
   Write a program to computerize the billing system of a hospital:

   - Design the class **doctorType** , inherited from the class personType , with an additional data member to store a doctors speciality. Add appropriate constructors and member functions to initialize, access, and manipulate the data members.
   - Design the class **billType** with data members to store a patients ID and a patients hospital charges, such as pharmacy charges for medicine, doctors fee, and room charges. Add appropriate constructors and member functions to initialize, access, and manipulate the data members.
   - Design the class **patientType**, inherited from the class personType, with additional data members to store a patients ID, age, date of birth, attending physicians name, the date when the patient was admitted in the hospital, and the date when the patient was discharged from the hospital. (Use the class dateType to store the date of birth, admit date, discharge date, and the class doctorType to store the attending physicians name.) Add appropriate constructors and member functions to initialize, access, and manipulate the data members.

   Write a program to test your classes.

3. *(**Hierarchical**)*: *When a single base class is used for the derivation of two or more classes, it is known as hierarchical inheritance.*
   An education institute wishes to maintain a database of its employees. The database is divided into a number of classes whose hierarchical relashionships are shown in the figure 1. The figure alse shows the minimum information required for each class. Specify all the classes and define functions to create the database and retrieve individual information as and when required.

4. *(**Multi-level**)*: *When a class is derived from another derived class, that is, the derived class acts as a base class, such a type of inheritance is known as multilevel inheritance.*

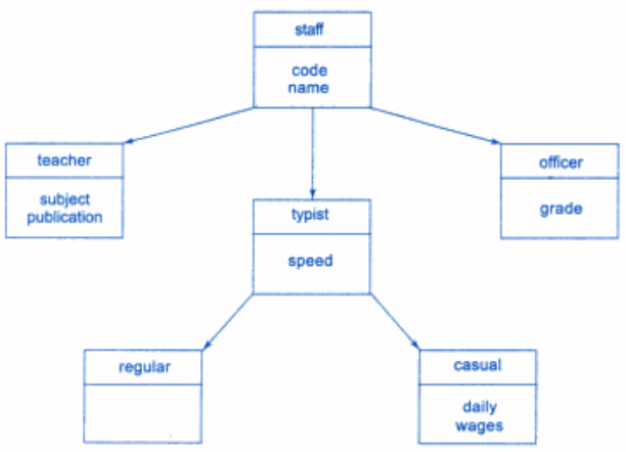

Figure 1: Hierarchical Inheritance

Discuss the multi-level inheritance scenario with reference to problem 3. (*write analytical discussion.*)

5. *(**Hybrid**):A combination of one or more types of inheritance is known as hybrid inheritance.*

   Discuss the hybrid inheritance scenario with reference to problem 3. (*write analytical discussion.*)

6. *(**Friend function**)*: A friend function is a function that can access the non-public members of a class,even though the function itself is not a member of the class. A class declaration must state which functions or classes will be its friends.

   (a) Create two classes. The first, named Sale, holds data for a sales transaction. Its private data members include the day of the month, amount of the sale, and the salespersons ID number. The second class, named Salesperson, holds data for a salesperson, and its private data members include each salespersons ID number and last name. Each class includes a constructor to which you can pass the field values. Create a friend function named display()that is a friend of both classes and displays the date of sale, the amount, and the salesperson ID and name. Write a short main()demonstration program to test your classes and friend function. Save the file as *Sales.cpp*.

   (b) Add a function to both the Sale and Salesperson classes that returns the private salesperson ID number. Write a main() function that contains an array of five Salesperson objects and store appropriate data in it. Then, continue to prompt the user for Sale data until the user enters an appropriate sentinel value. For each Sale transaction entered, determine whether the salespersons ID number is valid. Either display an error message, or use the friend display()function to display all the data. Save the file as *Sales2.cpp*.

Submission Details :

- Date of Issue : 13 - 02 - 2024
- Date of Submission : (Soft Copy, before 2 PM, 20-02-2024)
- Evaluation time : 2 - 5 PM