a

Object Oriented Paradigm

Lab Manual (Lab - 11)

Manual Contents:

- Inheritance
  - Rules of Inheritance
  - Access Specifiers
  - Multiple Inheritance and method over-riding

Session: Fall 2022

Faculty of Information Technology

UCP Lahore Pakistan

Objectives:

This activity has been designed so that you would be able to understand and implement following concepts:

- The basic concept of Inheritance

Instructions:

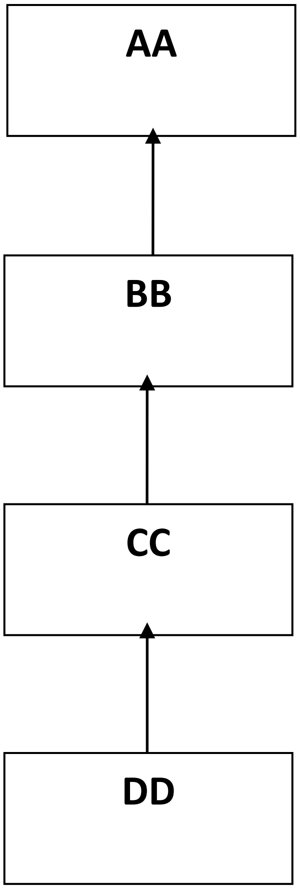
- Develop C++ programs for the following scenarios using OO design approach.
- Complete these problems in the lab.

Problems:

Task 01:

Order of Construction and Destruction

Implement the following scenario and show the execution order for construction and destruction of objects in inheritance hierarchy. Class AA is serving as the base class. Class BB is publicly inherited from class AA. Class is publicly inherited from class BB and finally the class DD is publicly inherited from class CC.



Task 02:

Base Initialization List

You have base class Person which has attributes:

- name: string
- age : int

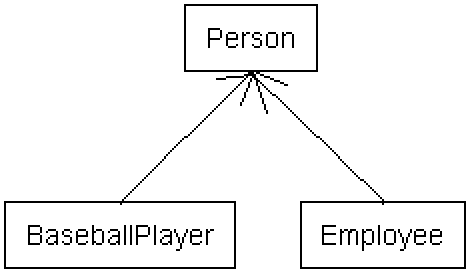
as its **private** member variables. Write parameterized constructor to initialize the values.

You have an Employee class which is publicly inherited from Person class. It has attributes:

- salary : double
- employeeId : int

You have another class BaseballPlayer publically inherited from Person class Baseball Player has attributes:

- battingAverage: double
- totalRuns: int



Make Baseball class object in main and initialize values of name, age, battingAverage and totalRuns using base initialization list.

Make Employee class object in main and initialize values of name, age, employeeId and salary using base initialization list.

Task 3:

Create a base class **Card** with the following attributes

- Card number : private
- Owner name : protected
- Expiry date : public

Derive the following classes from **Card**, with mentioned additional attributes

• **Calling card**(public inheritance)

- Amount : private
- Company name : private
- PIN : private

• **ID card**(protected Inheritance)

- CNIC Number : private
- Age : private

• **Driving license card**(private Inheritance)

- Driving license type (heavy, light, bike) : private
- Issued in city : private

Your tasks:

1. In the derived classes, write the getters and setters of every member variable(including the derived variables). You are not allowed to make any member function in the base class. A main() is required to add up some a card of each type, and then to display their information. The object of the base class will not be instantiated.
2. According to the rules of inheritance, clearly specify (by adding comments in the derived classes) which of the members are inherited and clearly mention their access specifiers in the derived classes.
3. Draw the UML diagram of the above mentioned scenario in any software of your choice.

TASK 4:

Create a base class, called BankAccount, and two additional classes (each derived from BankAccount), called SavingsAccount and CheckingAccount.

BankAccount:

- Title
- AccountNumber
- Balance
- Deposit()
- Withdraw()

SavingAccount:

- InterestRate
- CalculateInterest()

CheckingAccount:

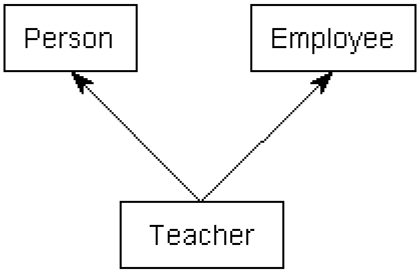
- fee charged per transaction

Class CheckingAccount should redefine member functions withdraw and deposit so that they subtract the fee from the account balance whenever either transaction is performed successfully.

You will then test the operations of each class in function main() to simulate the transactions of both a checking account and a savings account.

Task 05:

Provide the C++ implementation of the following.



The Person class has name and age as its attributes. It has an overloaded constructor to initialize the values and appropriate accessor and mutator methods. The Employee class has name of the employer and wage as its attributes with an overloaded constructor and appropriate accessor and mutator methods.

The Teacher class is inherited from Person and Employee class with an attribute of Pay Scale of type integer. It has overloaded constructor, appropriate accessor, mutator methods and a display function to print the Name, Age, Name of Employer, Wage and Pay Scale of Teacher.