Object Oriented Paradigm

Lab 11

Topic(s): Aggregation, Composition, Inheritance, Polymorphism, Static Data Members and Functions

---

IMPORTANT INSTRUCTIONS:

Please keep in mind the following points while coding. Violating any of these will result in credit deduction.

- There should be no memory leakage in your class. There should be no dangling pointers.
- Make functions, objects, variables as constant wherever possible.
- Create Default, Parameterized and Copy Constructor whether mentioned or not.
- Create Setters and Getters for all attributes.
- Follow the appropriate naming conventions as explained in class.

Question No. 01

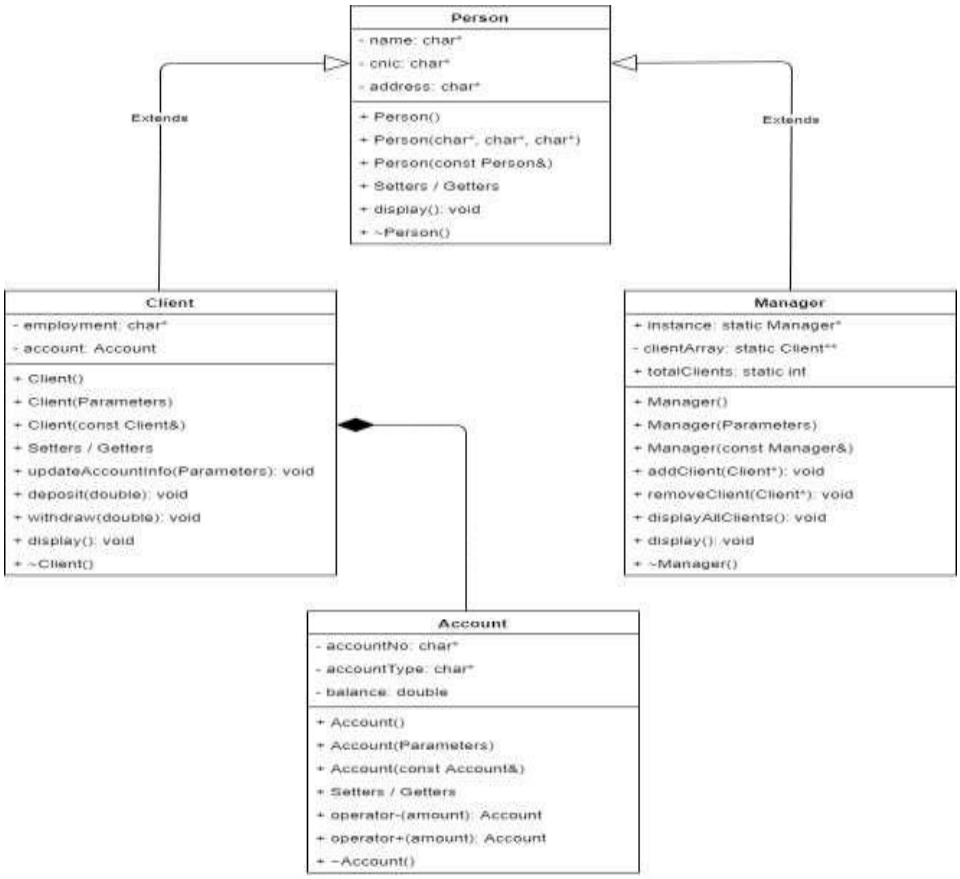
Create a class shape which has the following member attributes and functions.

- length
- area()
- perimeter()

Write a parameterized constructor to initialize the values. You have three derived classes Square, Rectangle and Trapezium. Each of these classes have different formulas for calculating the area and the perimeter. The formulas for calculating area and perimeter for each of these classes are given below. Each of the derived classes should redefine member functions for calculating area and perimeter as per the given table. Add additional attributes in the derived classes as per requirement.

	Area	Perimeter
Square	$A = l * l$	$P = 4l$
Rectangle	$A = l * w$	$P = 2\ (l + w)$
Trapezium	$A = \frac{(l + b)}{2} h$	$P = l + b + c + d$

Question No. 02



Create a class Person with the following attributes:

- name (char*)
- cnic (char*)
- address (char*)

Create another class Account with the following attributes and overloaded functions:

- accountNumber (char*)
- accountType – Current or Saving (char*)
- balance (double)
- Operator (+) to deposit money into the account.
- Operator (-) to withdraw money from the account.

Create a class Client with the following attributes and inherit it from the Person class:

- employment – Current Employment Title (char*)
- account (Account)

Now, do the following:

- Create a function that updates all the values in the client's account.
- Create deposit and withdraw functions each of which take a value as a parameter and call the appropriate operators for the account.

Create a class Manager with the following attributes and inherit it from the Person class:

- instance (static Manager*)
- clientArray (static Client**)
- totalClients (static int)

Now, do the following:

- Create an addClient function that take a client pointer as parameter and saves it in the clientArray by auto growing it.
- Create a removeClient function that take a client pointer as parameter and removes it from the clientArray and shrinks the array.
- Create a displayAllClients function that prints all the clients the manager is dealing with.

CHALLENGE: There is only one Manager! No other Manager can be created!