National University of Computer and Emerging Sciences



Lab Manual # 11

Object Oriented Programming

(CL2004)

|  |  |
| --- | --- |
| Course Instructor | Ms. Hafsa Tariq |
| Lab Instructor (s) | Sonia Anum  Yusra Arshad |
| Section | 2J |
| Semester | Spring 2022 |

Department of Computer Science

FAST-NU, Lahore, Pakistan

## Objectives

After performing this lab, students shall be able to understand:

* Inheritance and its types

**TASK 1:**

Consider a base class named Employee and its derived classes **HourlyEmployee** and **PermanentEmployee** while taking into account the following criteria.

* ***Employee*** class has two data fields i.e. a ***name*** (of type string) and specific ***empID*** (of type integer)
* Both classes (***HourlyEmployee*** and ***PermanentEmployee***) have an attribute named ***hourlyIncome***
* Both classes (***HourlyEmployee*** and ***PermanentEmployee***) have ***three-argument*** constructor to initialize the ***hourlyIncome*** as well as data fields of the base class
* Class ***HourlyEmployee*** has a function named ***calculate\_the\_hourly\_income*** to calculate the income of an employee for the actual number of hours he or she worked. One hour income is Rs. 150
* Similarly, ***PermanentEmployee*** class has function named ***calculate\_the\_income*** to calculate the income of an employee that gets paid the salary for exact 240 hours, no matter how many actual hours he or she worked. Again, one hour salary is ***Rs. 150.***

Implement all class definitions with their respective ***constructors*** to initialize all data members and functions to compute the total income of an employee. In the **main**() function, create an instance of both classes (i.e. ***HourlyEmployee*** and ***PermanentEmployee***) and test the working of functions that calculate total income of an employee.

**Task 2:**

Consider a class ***BankAccount*** that has

* Two attributes i.e. ***accountID*** and balance and
* A function named ***balanceInquiry***() to get information about the current amount in the account Derive two classes from the ***BankAccount*** class i.e. ***CurrentAccount*** and the ***SavingsAccount***. Both classes (***CurrentAccount*** and ***SavingsAccount***) inherit all attributes/behaviors from the ***BankAccount*** class. In addition, followings are required to be the part of both classes
* Appropriate ***constructors*** to initialize data fields of base class
* A function named ***amountWithdrawn***(amount) to withdraw certain amount while taken into account the following conditions
* While withdrawing from current account, the minimum balance should not decrease ***Rs. 5000***
* While withdrawing from savings account, the minimum balance should not decrease ***Rs. 10,000***
* ***amountDeposit(amount)*** to deposit amount in the account

In the **main()** function, create instances of derived classes (i.e. ***CurrentAccount*** and ***SavingsAccount***) and invoke their respective functions to test their working.

**Task 3:**

Consider the following details of all classes for class hierarchy.

• Class ***Person*** holds

* Two attributes i.e. ***name*** and ***year\_of\_birth***
* A ***two-argument constructor*** to initialize its data members with user-defined values

• Class ***Student*** has

* Two attributes i.e. studentID and enrolledSemester
* A ***four-argument constructor*** to initialize its data members (including inherited data members)
* A function named ***display()*** to show the values of all attributes (including inherited attributes)

• Class ***Employee*** contains

* Five attributes i.e. ***employeeID, joiningYear, jobTitle*** (designation of an employee), ***courseID***, and ***courseTitle.***

• Class ***Administration*** has

* A ***parameterized constructor*** to receive ***five arguments*** to initialize inherited attributes from class ***Employee*** (Concerning ***courseID*** and ***courseTitle***, only null value is allowed to set for an admin officer)
* Two functions i.e. ***setJobTitle***(employee) and ***getJobTitle***(employee) to set and get job title of an employee.

• Class ***Academic*** has

* A ***parameterized constructor*** to receive ***five arguments*** to initialize inherited attributes from class Employee (Concerning ***employeeID, joiningYear, and jobTitle***, only null value is allowed to set)
* Two functions i.e. ***setCourseID()*** and ***setCourseTitle()***

• Only an instance of class ***DeanHOD*** should be able to modify values for ***employeeID, designation of an employee, ID*** and ***name of a particular course.***

Implement all these classes and within the main function, create instances of all classes (except class Employee) and test the described working of all these classes.