University of Central Punjab

**Faculty of Information Technology**

# Object Oriented Programming

# Spring 2023

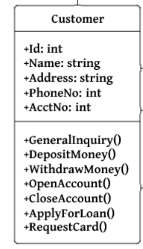
|  |  |  |
| --- | --- | --- |
| **Lab 03** | |  |
| **Topic** | classes in C++, UML diagram, templates |
| **Objective** | * Separate Interface from implementation and driver (working in 3 different files) * Good Code Writing Style and Class Templates * Simple Classes with attributes and simple functions and Main Driver * UML Class Diagram * Dry Run and use of the Debugger to look at variable addresses and values |
|  | | |

**Instructions:**

* Indent your code.
* Comment your code.
* Use meaningful variable names.
* Plan your code carefully on a piece of paper before you implement it.
* Name of the program should be same as the task name. i.e. the first program should be Task\_1.cpp
* **void main() is not allowed. Use int main()**
* Implement it in multiple files. Every task will contain three respective files Class.h Class.cpp and task.cpp (main().cpp)
* **You are not allowed to use any built-in functions**
* **You are required to follow the naming conventions as follow:**
  + **Variables:** firstName; (no underscores allowed)
  + **Function:** getName(); (no underscores allowed)
  + **ClassName:** BankAccount (no underscores allowed)

**Students are required to complete the following tasks in lab timings.**

**Task 1:**





Keeping in view the above UML class diagram write the necessary code by using three different files.

**Task 2:**

Using the concept of getters and setters, Create a class with name Employee that contains a private data member with the name **salary** and take the salary of the employee from the user in the main() function and set that salary into the data member of class and the name of the data member is salary, you have to use a function setSalary(int s); and then get the salary back in the main function using a getter function with a name getSalary();

**Task 3:**

Create a class named as **Employee** having following private attributes:

## employeeCode(int)

1. **ageAtJoining(int)**
2. **currentAge(int)**

Now write the following for the above mentioned class:

* Write a function **constructor** which should receive three integers as parameters (\_employeeCode, \_ageAtJoining and \_currentAge) and initialize all the attributes with the respective values received in the parameters.

**Remember:** Validate the values in the above function. For eg: ageAtJoining can never be negative etc.

* Write a non-returning **display** function to print the attributes of the class.
* Write a function **calculateTenure** which calculates the tenure of the employee and returns it.

Write a program to create five objects of employee with different data. Display the details of those employees whose tenure is 2 or more than 2 years.

**Class templates**

