# Project

Only Customers can open accounts

# OOP Concepts

Object Oriented Programming (OOP) is a programming paradigm in which problems are modelled as real world objects. Following sections explain how different aspects of the OOP are implemented to this project.

## Classes

Classes are blue prints for creating objects and they contains the information on what properties (attributes) and behaviours (methods) its instances (objects) possess. Following are some classes of this project:

\*User

\*Customer

\*Employee

\*Account

Picture of a class code

## Attributes

Attributes are the properties of an object. For instance, fname (first name) is an attribute of the Customer class and its objects.

Picture of attribute code

## Methods

Methods are the behaviours of an object. For instance, balance check is an attribute of the Account class and its objects.

Picture of a method code.

## Constructuors

Constructors are methods

## Inheritence

## Polymorphism

Polymorhpism

## Abstraction

User class is an abstract class of employee and custoemr class

## Encapsulation

Most methods are private or private which are given access to useing getter and setter methods

No getter for password

# Data Structures

HashMaps in employee class get custoemr informaitn method.

# SOLID Design Principles

## Single Responsibility

Each class serve one purpose

## Open Closed

Output class only prints the information given to it. You don’t have to change the class to accomodate the different types of accounts or custoemrs.

## Liskov Substitution

Even thoug ther are some additional methods in the employee and customer class, their base class (user) does not force any un implementable behaviour on them.

## Interface Segregation

There is user class and IemployeeUser interface.

## Dependency Inversion Principal

Account instances are created from Iaccount interface not the account class