**Assignment 2: Classes, Objects, and Methods**

**Objective: Develop a deeper understanding object-oriented programming principles.**

**Creating Class Person:**

package BankAccount;

public class Person {

private String name;

private int age;

public Person(String name, int age) {

this.name = name;

this.age = age;

}

public String getName() {

return name;

}

public int getAge() {

return age;

}

public void displayName() {

System.*out*.println("Name: " + name);

}

public void displayAge() {

System.*out*.println("Age: " + age);

}

public void displayInfo() {

System.*out*.println("Name: " + name + ", Age: " + age);

}

}

**Creating a Class Bank Account:**

package BankAccount;

import java.util.\*;

public class BankAccount{

private String accountHolder;

private double balance;

// Constructor

public BankAccount(String accountHolder, double initialBalance) {

this.accountHolder = accountHolder;

this.balance = initialBalance;

}

public void deposit(double amount) {

if (amount > 0) {

balance += amount;

System.*out*.println("Deposited $" + String.*format*("%.2f", amount) + ". New balance: $" + String.*format*("%.2f", balance) + ".");

} else {

System.*out*.println("Deposit amount must be positive.");

}

}

public void withdraw(double amount) {

if (amount > 0) {

if (amount <= balance) {

balance -= amount;

System.*out*.println("Withdrew $" + String.*format*("%.2f", amount) + ". New balance: $" + String.*format*("%.2f", balance) + ".");

} else {

System.*out*.println("Insufficient funds.");

}

} else {

System.*out*.println("Withdrawal amount must be positive.");

}

}

public void displayBalance() {

System.*out*.println("Account balance: $" + String.*format*("%.2f", balance));

}

}

Creating a Class Main:

**package** BankAccount;

**public** **class** **Main** {

**public** **static** **void** **main**(**String**[] args) {

**Person** **person** = **new** Person("Latha", 21);

person.displayName();

person.displayAge();

person.displayInfo();

**BankAccount** **account** = **new** BankAccount(person.getName(), 1000);

account.displayBalance();

account.deposit(200);

account.withdraw(400);

account.withdraw(1100);

account.displayBalance();

}

}