Task 2:

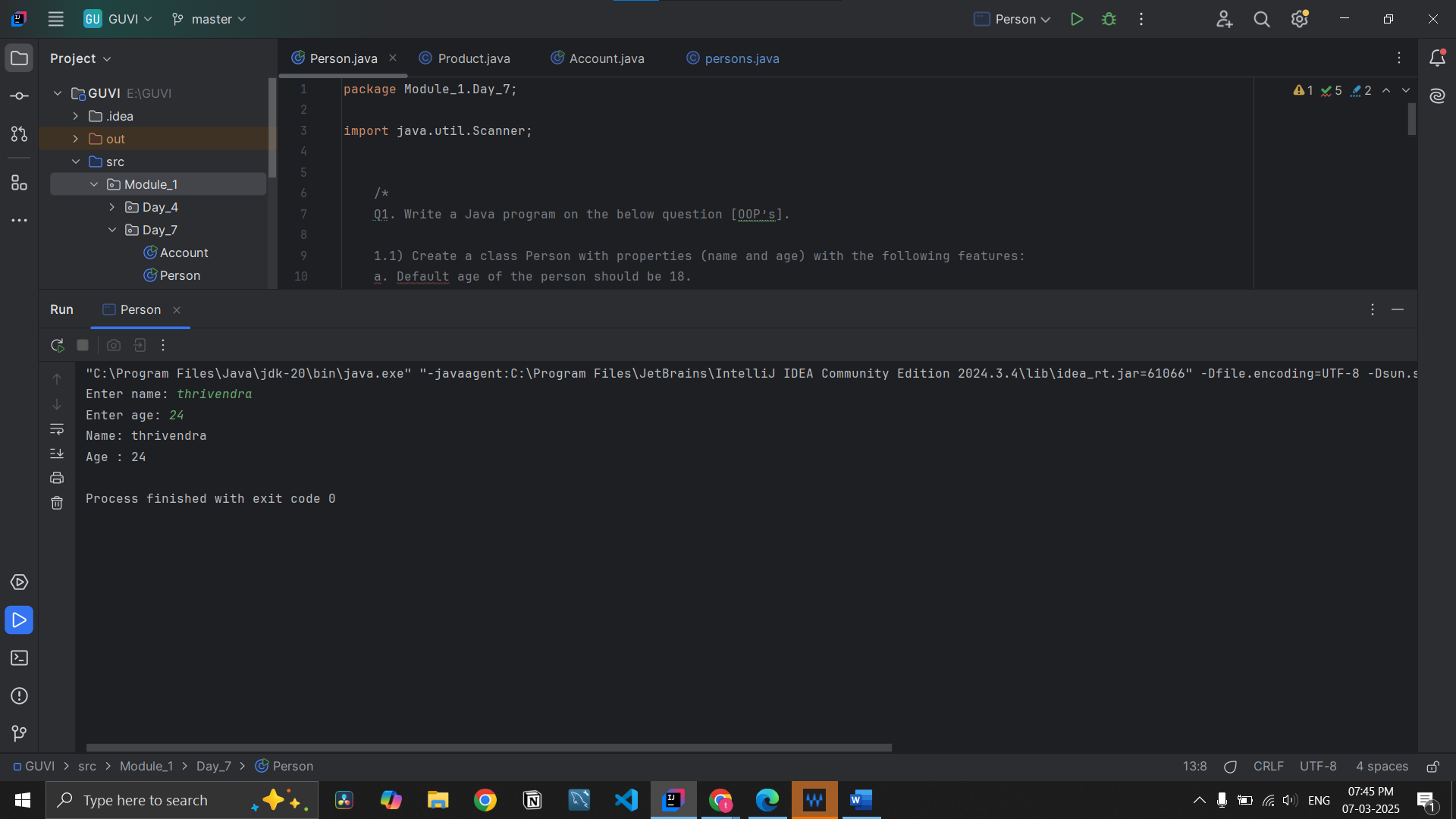
1.1) Create a class Person with properties (name and age) with the following features:

a. Default age of the person should be 18.  
b. A Person object can be initialized with name and age.  
c. Method to display name and age of the person.

**Code:**

package Module\_1.Day\_7;  
  
import java.util.Scanner;  
  
  
 /\*  
 Q1. Write a Java program on the below question [OOP’s].  
  
 1.1) Create a class Person with properties (name and age) with the following features:  
 a. Default age of the person should be 18.  
 b. A Person object can be initialized with name and age.  
 c. Method to display name and age of the person.  
 \*/  
public class Person  
{  
 // Creating two variables to store the data of the object  
 public String name;  
 public int age=18;  
  
 //constructor to initialize name and age  
 public Person(String name, int age)  
 {  
 // The 'this' keyword is used to differentiate between instance variables (global) and local variables.  
 this.name = name;  
 this.age = age; // The Default age is set to 18  
 }  
  
 // Print the data using display method  
 public void display()  
 {  
 System.*out*.println("Name: "+this.name);  
 System.*out*.println("Age : "+this.age);  
 }  
  
 public static void main(String[] args) {  
  
 Scanner input = new Scanner(System.*in*);  
 System.*out*.print("Enter name: ");  
 String userName = input.nextLine();  
 System.*out*.print("Enter age: ");  
 int userAge = input.nextInt();  
  
 // Creating an object for Person class  
 Person userPerson = new Person(userName, userAge);  
 userPerson.display();  
  
 }  
}

Output screenshot:

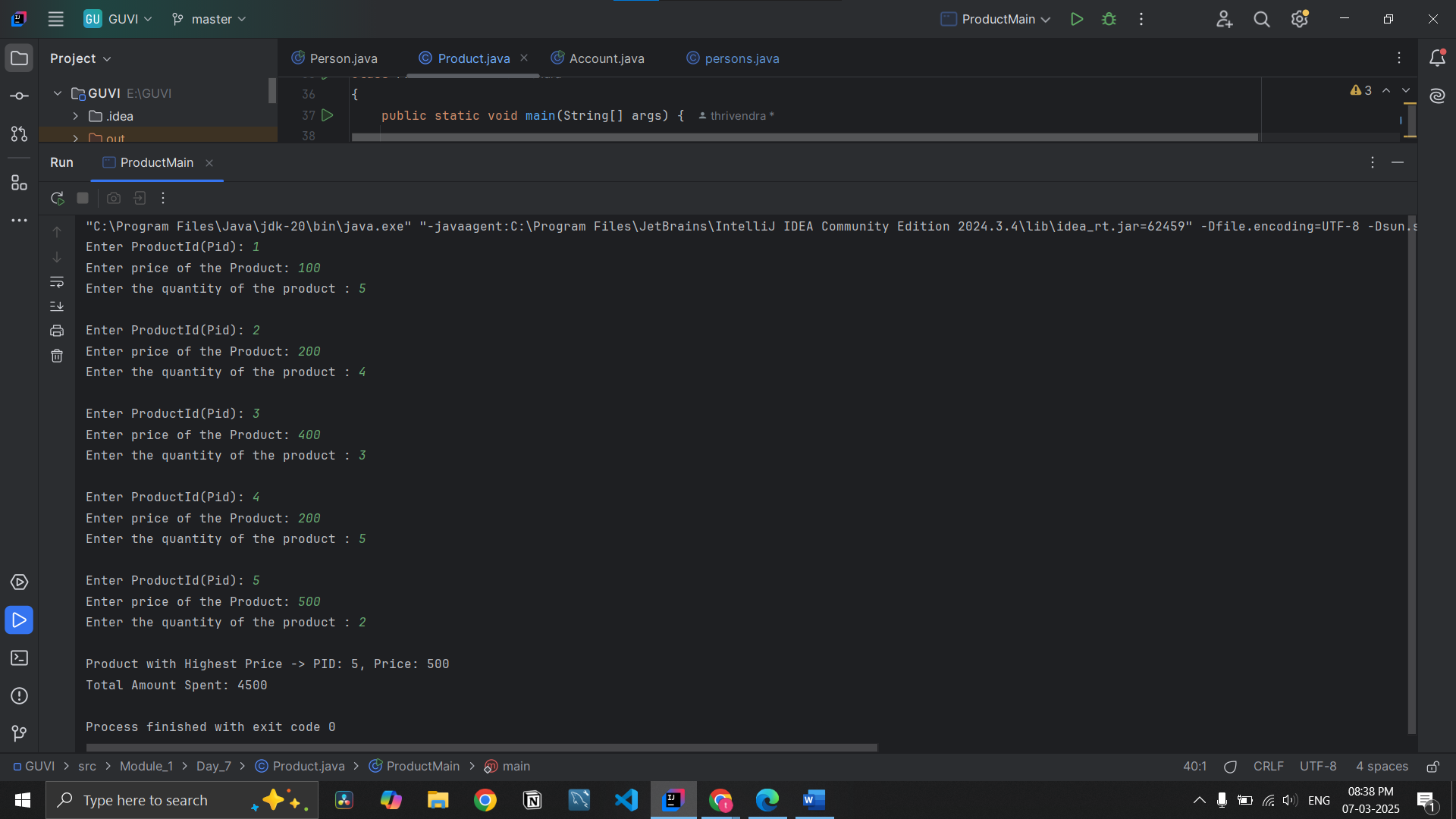


**1.2)** Create a class Product (pid, price, quantity) with a parameterized constructor.  
Create a main function in a different class (say ProductMain) and perform the following tasks:  
  
a. Accept five product information from the user and store them in an array.  
B. Find the pid of the product with the highest price.  
c. Create a method (with an array of product objects as an argument) in the ProductMain class to calculate and return the total amount spent on all products.  
(Amount spent on a single product = price of product × quantity of product).

Code:

package Module\_1.Day\_7;  
  
import javax.swing.\*;  
import java.util.Scanner;  
  
/\*  
1.2) Create a class Product (pid, price, quantity) with a parameterized constructor.  
Create a main function in a different class (say ProductMain) and perform the following tasks:  
  
a. Accept five product information from the user and store them in an array.  
B. Find the pid of the product with the highest price.  
c. Create a method (with an array of product objects as an argument) in the ProductMain class to calculate and return the total amount spent on all products.  
(Amount spent on a single product = price of product × quantity of product).  
 \*/  
  
  
  
// Product class to store product details (PID, price, and quantity)  
public class Product  
{  
 public int pid; // Product ID  
 public long price; // Product Price  
 public int quantity; // Product Quantity  
  
 // Parameterized constructor to initialize product attributes  
 public Product(int pid, long price, int quantity) {  
 this.pid = pid;  
 this.price = price;  
 this.quantity = quantity;  
 }  
  
}  
  
// Main class to handle input, process data, and display results  
class ProductMain  
{  
 public static void main(String[] args) {  
  
 Scanner input=new Scanner(System.*in*);  
  
 // Creating an array to store 5 products  
 Product[] product=new Product[5];  
  
 // Loop to accept product details from the user  
 for (int i = 0; i < product.length; i++)  
 {  
 System.*out*.print("Enter pid: ");  
 int pid=Integer.*parseInt*(input.nextLine());  
  
 System.*out*.print("Enter price: ");  
 long price=Long.*parseLong*(input.nextLine());  
  
 System.*out*.print("Enter the quantity between : ");  
 int quantity=Integer.*parseInt*(input.nextLine());  
  
 product[i]=new Product(pid,price,quantity);  
 System.*out*.println();  
 }  
  
 // Creating an object of ProductMain to call non-static methods  
 ProductMain productMain=new ProductMain();  
  
  
 // Finding the highest price product  
 Product highest = productMain.HighestPrice(product);  
 System.*out*.println("Product with Highest Price -> PID: " + highest.pid + ", Price: " + highest.price);  
  
 // Calculating total amount spent  
 long totalAmount = productMain.Total\_Product\_prices(product);  
 System.*out*.println("Total Amount Spent: " + totalAmount);  
  
 }  
  
 // Method to find the product with the highest price  
 public Product HighestPrice(Product[] product)  
 {  
  
 Product maxproduct=product[0];  
  
  
 for (int i = 1; i < product.length; i++)  
 {  
 if(product[i].price>maxproduct.price)  
 {  
 maxproduct=product[i]; // Update maxProduct if a higher price is found  
 }  
  
 }  
 return maxproduct;  
  
 }  
  
 // Method to calculate the total amount spent on all products  
  
 public long Total\_Product\_prices(Product[] products)  
 {  
 long totalPrice=0;  
  
 for (int i = 0; i < products.length; i++)  
 {  
 totalPrice+= (long) products[i].price \*products[i].quantity;  
  
  
 }  
 return totalPrice;  
  
 }  
}

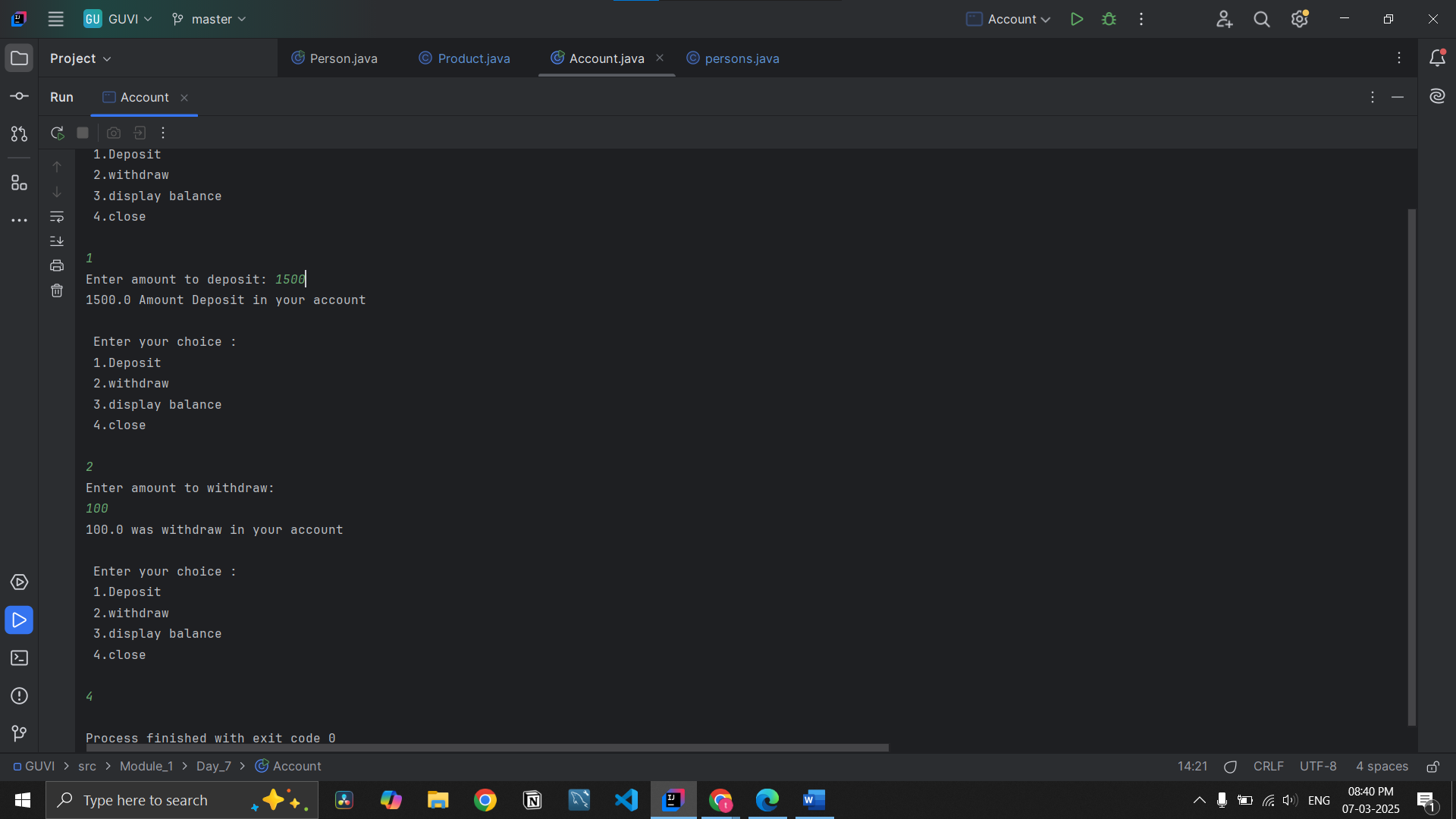
Output screenshot:



**1.3)** Create a Class Account with data member Balance.  
Create two constructors (one with no argument and one with an argument) and perform the following tasks:  
  
a. Method to deposit the amount into the account.  
b. Method to withdraw the amount from the account.  
c. Method to display the balance.

**Code:**  
  
package Module\_1.Day\_7;  
  
import java.util.Scanner;  
  
  
/\*  
1.3) Create a Class Account with data member Balance.  
Create two constructors (one with no argument and one with an argument) and perform the following tasks:  
  
a. Method to deposit the amount into the account.  
b. Method to withdraw the amount from the account.  
c. Method to display the balance.  
 \*/  
public class Account  
{  
 private double balance; // Data member to store  
  
 // Default constructor (initializes balance to 0)  
 public Account()  
 {  
 this.balance=0.0;  
 }  
  
 // Parameterized constructor (initializes balance with a given amount)  
  
 public Account(double balance) {  
 this.balance = balance;  
 }  
  
 // Method to deposit amount into the account  
  
 public void deposit(double amount)  
 {  
 if(amount>0)  
 {  
 balance+=amount;  
 System.*out*.println(amount+" Amount Deposit in your account");  
 }  
 else {  
 System.*out*.println("Invalid deposit amount");  
 }  
  
 }  
  
 // Method to withdraw amount from the account  
  
 public void withdraw(double amount)  
 {  
 if(amount>0 && amount<=balance)  
 {  
 balance-=amount;  
 System.*out*.println(amount+" was withdraw in your account");  
 }  
 else {  
 System.*out*.println("Insufficient balance or invalid amount!");  
 }  
  
 }  
  
 // Method to display the account balance  
  
 public void displaybalance()  
 {  
 System.*out*.println("Total amount in your account "+balance);  
 }  
  
 public static void main(String[] args)  
 {  
 Scanner input=new Scanner(System.*in*);  
  
 // Creating an object of Account class  
 Account account=new Account();  
  
  
 boolean flag=true;  
 // Using a while loop for continuous user interaction until exit  
 while (flag)  
 {  
 System.*out*.println(" \n Enter your choice :" +  
 "\n 1.Deposit " +  
 "\n 2.withdraw" +  
 "\n 3.display balance " +  
 "\n 4.close \n");  
 int choice = input.nextInt();  
  
 // Using switch-case to execute correct functions based on user input  
 switch (choice)  
 {  
 case 1: {  
 System.*out*.print("Enter amount to deposit: ");  
 double amount = input.nextDouble();  
 account.deposit(amount);  
 break;  
 }  
  
 case 2:  
 {  
 System.*out*.println("Enter amount to withdraw: ");  
 double amount=input.nextDouble();  
 account.withdraw(amount);  
 break;  
 }  
  
 case 3:  
 {  
 account.displaybalance();  
 break;  
 }  
 case 4:  
 {  
 flag=false;  
 break;  
 }  
 default:  
 {  
 System.*out*.println("Invalid choice! Please try again.");  
 }  
 }  
  
 }  
  
 }  
}

**Output:**



**1.4)** Define a base class Person with attribute name and age:  
Create a subclass Employee that inherits from Person and adds attributes like employeeID and salary.  
Use the super keyword to initialize the Person attributes in the Employee constructor.  
 \*/  
// Employee class that inherits from Person

**Code:**

package Module\_1.Day\_7;  
  
import java.util.Scanner;  
  
/\*  
1.4) Define a base class Person with attribute name and age:  
Create a subclass Employee that inherits from Person and adds attributes like employeeID and salary.  
Use the super keyword to initialize the Person attributes in the Employee constructor.  
 \*/  
// Employee class that inherits from Person  
  
public class persons {  
 // Creating two variables to store the data of the object  
 public String name;  
 public int age ;  
  
 //constructor to initialize name and age  
 public persons(String name, int age) {  
 // The 'this' keyword is used to differentiate between instance variables (global) and local variables.  
 this.name = name;  
 this.age = age; // The Default age is set to 18  
 }  
  
 // Print the data using display method  
 public void display() {  
 System.*out*.println("Name: " + this.name);  
 System.*out*.println("Age : " + this.age);  
 }  
  
}  
 class Employee extends persons  
{  
 public int employeeID;  
 public double salary;  
  
 // Constructor to initialize Employee attributes using super() for name and age  
  
 public Employee(String name, int age,int employeeID,double salary) {  
 super(name, age); // Calling the constructor of the Person class  
 this.employeeID=employeeID;  
 this.salary=salary;  
 }  
  
 // Method to display employee details  
 public void displayemployee()  
 {  
 super.display(); // Displaying Person (name & age) details  
 System.*out*.println("EmployeeId: "+employeeID+" \nSalary: "+salary);  
 }  
  
 // Main method to take user input and display Employee details  
  
 public static void main(String[] args) {  
 Scanner input = new Scanner(System.*in*);  
  
 // Taking input from the user  
 System.*out*.print("Enter name: ");  
 String userName = input.nextLine();  
  
 System.*out*.print("Enter age: ");  
 int userAge = input.nextInt();  
  
 System.*out*.print("Enter EmployeeId: ");  
 int employeeId=input.nextInt();  
  
 System.*out*.print("Enter salary: ");  
 double salary=input.nextDouble();  
 // Creating an Employee object with user input  
 Employee employee=new Employee(userName,userAge,employeeId,salary);  
 // Displaying Employee details  
 employee.displayemployee();  
 }  
}

**output Screenshot**

