GUVI Task – II

Introduction to Oops in java.

1) Q1. write a java program on below questions [OOP's].

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

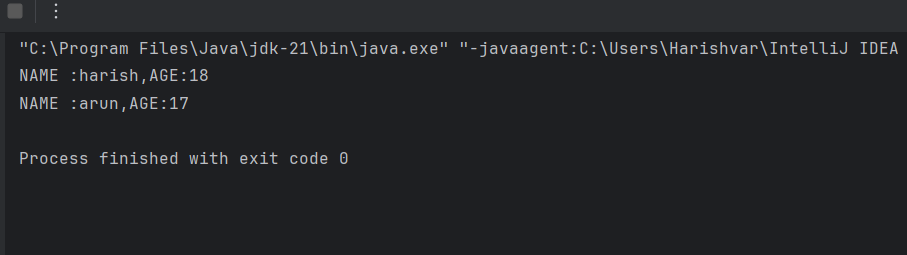
a. Default age of person should be 18;

b. A person object can be initialized with name and age;

c. Method to display name and age of person

package OOPS;  
  
public class Person {  
 private int age = 18;  
 private String name;  
 //Default of age of person is initialized as 18.  
 public Person(String name){  
 this.name = name;  
 this.age=age;  
  
 }  
  
 // Initializing name and age with the person object.  
 public Person(String name,int age) {  
 this.name = name;  
 this.age = age;  
 }  
  
 // Function to display the Person details name and age.  
 public void display(){  
 System.*out*.println("NAME :"+name+",AGE:"+age);  
  
 }  
 public static void main(String[] args){  
 Person person1 = new Person("harish");  
 person1.display();  
  
 Person person2 = new Person("arun",17);  
 person2.display();  
 }  
   
}

Output:



2) Create Class Account with data member as Balance. Create two constructors (no argument, and with argument) and perform following task

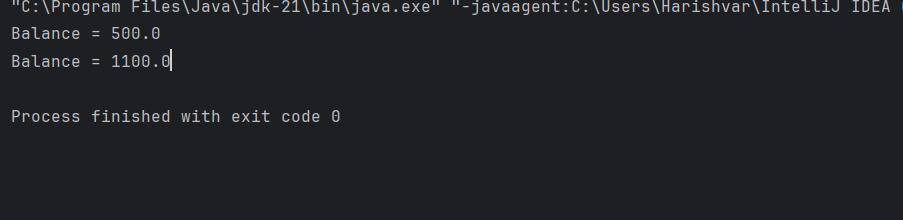
a. method to deposit the amount to the account.

b. method to withdraw the amount from the account.

C. method to display the Balance

package OOPS;  
  
public class Account {  
 private double balance;  
  
 //constructor with no argument & default balance.  
 public Account(){  
 this.balance = 0.0;  
 }  
 //Constructor with argument   
 public Account(double initialBalance){  
 balance += initialBalance;  
 }  
  
  
 //method to Deposite the amount to the account.  
 public void deposite(int depositeAmount)  
 {  
 this.balance += depositeAmount;  
 }  
  
  
 //method to Withdraw the amount from the account.  
 public void withdraw(int withdrawAmount){  
 this.balance -= withdrawAmount;  
 }  
  
  
 //method to display Balance  
 public void displayBalance(){  
 System.*out*.println("Balance = "+ balance);  
 }  
 public static void main(String[] args) {  
 Account acc = new Account();  
 acc.deposite(1000);  
 acc.withdraw(500);  
 acc.displayBalance();  
  
 Account acc1 = new Account(1000);  
 acc1.deposite(200);  
 acc1.withdraw(100);  
 acc1.displayBalance();  
  
  
 }  
 }

Output:



3) Method to display name and age of person

1.2). Create class Product (pid, price, quantity) with parameterized constructor. Create a main function in different class (say ProductMain) and perform following task:

a. Accept five product information from user and store in an array

b. Find Pid of the product with the highest price.

c. Create method (with array of product's object as argument) in ProductMain class to calculate and return the total amount spent on all products. (amount spent on single product-price of product \* quantity of product.

4)

package OOPS;  
  
public class Person1 {  
 protected String name;  
 protected int age;  
  
 public Person1(String name,int age) {  
 this.name = name;  
 this.age = age;  
 }  
  
 public void displayPersonInfo() {  
 System.*out*.println("Name: " + name);  
 System.*out*.println("Age: " + age);  
 }  
}  
  
class Employee extends Person1{  
 long EmployeeId;  
 int salary;  
  
 public Employee(String name,int age,long EmployeeId,int salary) {  
 super(name,age);  
 this.EmployeeId = EmployeeId;  
 this.salary = salary;  
 }  
  
 public static void main(String[] args) {  
 Employee emp = new Employee("harish",21,1235678910L,20000);  
 emp.displayPersonInfo();  
 }  
}

Output:

