1.1) create a class person with properties (name and age) with following features.

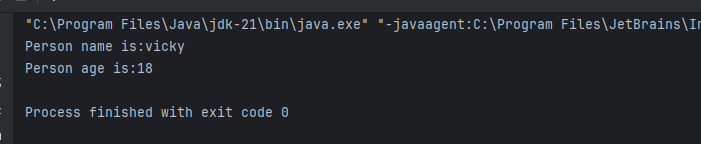
a) Default age should be 18

b) A person object can be initialized with name and age

c) method to display name and age of person

public class Person {  
  
 private String name;  
 //Setting defalut age as 18  
 final private int age=18;  
  
 //constructor to initialize the object with parameters  
 public Person(String name,int age)  
 {  
 this.name=name;  
 }  
 //Method to display person name and age  
 public void displayPersonInfo()  
 {  
 System.*out*.println("Person name is:"+name+"\n"+"Person age is:"+age);  
 }  
  
 //Main method to access person class method to display person details  
 public static void main(String[] args)  
 {  
 Person person=new Person("vicky",20);  
 person.displayPersonInfo();  
  
 }  
}

Output:

****

**1.2 create a class product (pid,price,quantity) with parameterized constructor. create a main function in different class(say ProductMain) and perfrom following task**

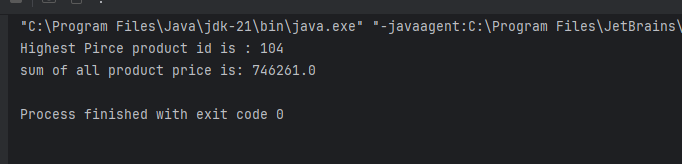
**a)Accept five product information from user and store it in array**

**b) Find pid of the product with highest price**

**c) create method (with array of product 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 the product)**

public class Product {  
  
 private int pid;  
 private double price;  
 private int quantity;  
  
 //parameterized constructor for product object  
 public Product(int pid, double price, int quantity) {  
 this.pid = pid;  
 this.price = price;  
 this.quantity = quantity;  
 }  
 public double getPrice() {  
 return price;  
 }  
 public int getPid() {  
 return pid;  
 }  
 public int getQuantity() {  
 return quantity;  
 }  
}  
class ProductMain  
{  
 public static void main(String[] args)  
 {  
 double highestPrice=Double.*MIN\_VALUE*;  
 int productId=0;  
 //creating a array to store product details  
 Product[] prdArray=new Product[5];  
 prdArray[0]=new Product(101,14000.00,4);  
 prdArray[1]=new Product(102,12500.00,6);  
 prdArray[2]=new Product(103,1000.50,19);  
 prdArray[3]=new Product(104,190000.50,3);  
 prdArray[4]=new Product(105,1050.00,25);  
  
 //iteration to find out the highest product  
 for(int i=0;i<=prdArray.length-1;i++)  
 {  
 if(highestPrice <= prdArray[i].getPrice())  
 {  
 highestPrice=prdArray[i].getPrice();  
 productId=prdArray[i].getPid();  
 }  
 }  
 System.*out*.println("Highest Pirce product id is : "+productId);  
 //calling function to calculate total amount spent on all produts  
 *calculateTotalAmount*(prdArray);  
 }  
  
 // method to find out total amount  
 public static void calculateTotalAmount(Product[] prdArray)  
 {  
 double eachPrdPrice=0.00;  
 double sumOfAllPrdPrice=0.00;  
 for(int i=0;i<=prdArray.length-1;i++)  
 {  
 eachPrdPrice=prdArray[i].getPrice()\*prdArray[i].getQuantity();  
 sumOfAllPrdPrice=sumOfAllPrdPrice+eachPrdPrice;  
 }  
 System.*out*.println(sumOfAllPrdPrice);  
 }  
}

**output:**

****

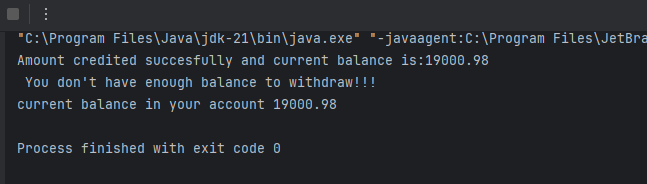
**1.3 create a class Account with data member as balance .create two constructors(no arguments,with arguments) and perfrom follwing task**

**a. Method to deposit amount**

**b. Method to withdraw the amount**

**c. Method to display balance**

public class Account {  
  
 private double balance;  
  
 //no argument constructor  
 public Account() {  
 }  
  
 //Parameterized constructor  
 public Account(double balance) {  
 this.balance = balance;  
 }  
  
 //Method to deposit amount in account  
 public void depositAmount(double amount)  
 {  
 balance=balance+amount;  
 System.*out*.println("Amount credited succesfully and current balance is:"+balance);  
 }  
  
 //Method to withdraw the amount from balance  
 public void withdrawAmount(double amount)  
 {  
 if(balance >= amount)  
 {  
 balance=balance-amount;  
 System.*out*.println("Amount debited succesfully and current balance is:"+balance);  
 }  
 else {  
 System.*out*.println(" You don't have enough balance to withdraw!!!");  
 }  
 }  
  
 //Method to display the account balance  
 public void displayBalance()  
 {  
 System.*out*.println("current balance in your account "+this.balance);  
 }  
  
 //Main method to invoke the function  
 public static void main(String[] args)  
 {  
  
 Account acc=new Account();  
 acc.depositAmount(19000.98);  
 //acc.withdrawAmount(4530);  
 acc.withdrawAmount(9063242.00);  
 acc.displayBalance();  
 }  
}

****

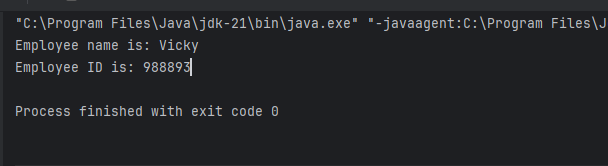
**1.4 Define a base class person with attributes name and age**

**create a subclass Employee that inherits from Person and adds attributes like empolyeeID and salary**

**Use the super keyword to initialize the person attributes in the Employee constructor**

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;  
 }  
}  
class Employee extends Person  
{  
 private int employeeID;  
 private double salary;  
  
 public Employee(String name, int age, int employeeID, double salary) {  
 super(name, age);  
 this.employeeID = employeeID;  
 this.salary = salary;  
 }  
 public int getEmployeeID() {  
 return employeeID;  
 }  
  
 public static void main(String[] args)  
 {  
 Employee emp=new Employee("Vicky",20,988893,900700);  
 System.*out*.println("Employee name is: "+emp.getName()+"\n"+"Employee ID is: "+emp.getEmployeeID());  
 }  
 }

**output:**

****