**1.Create a class Person with properties (name and age)**

**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**

**CODE**

**// Define a class named Person**

public class Person1 {

   // Declare variables

   String name;

   int age = 18; // Age variable default value of 18

   // Method to display the name and age

   void display() {

      System.out.println(name);

      System.out.println(age);

   }

   public static void main(String[] args) {

     // Create an object of the Person class

     Person1 a = new Person1();

     // Assign name using object

     a.name = "suhail";

     // Call the display method using object

     a.display();

   }

}

**OUTPUT**

**A black background with white text

AI-generated content may be incorrect.**

**2. Create class Product (pid, price, quantity) with parameterized constructor.**

**create a main function in different class 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 in ProductMain class to calculate and return the total amount spent on all products.**

**CODE**

import java.util.Scanner;

class Product {

    int pid;      // Product ID

    double price; // Product Price

    int quantity; // Quantity of Product

    // Parameterized constructor to initialize product details

    public Product(int pid, double price, int quantity) {

        this.pid = pid;

        this.price = price;

        this.quantity = quantity;

    }

}

public class ProductMain {

    // Method to calculate the total amount spent on products

    public static void calculateTotalAmount(Product[] products) {

        double totalAmount = 0;

        for (int i = 0; i < 5; i++) {

            totalAmount = totalAmount + (products[i].price \* products[i].quantity);

        }

        System.out.println("Total Amount Spent: " + totalAmount);

    }

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in); // Scanner object for user input

        Product[] products = new Product[5]; // Array to store 5 product objects

        // Loop to take product details from user

        for (int i = 0; i < 5; i++) {

        System.out.println("Enter Product ID, Price, and Quantity for Product " + (i + 1) + ":");

            int pid = scanner.nextInt();

            double price = scanner.nextDouble();

            int quantity = scanner.nextInt();

            products[i] = new Product(pid, price, quantity);

        // Find the product ID with the highest price

        int highestPricePid = products[0].pid; // Assume first product has highest price

        double highestPrice = products[0].price;

        // Loop through the product list to find the highest price

        for (int i = 1; i < products.length; i++) {

            if (products[i].price > highestPrice) {

                highestPrice = products[i].price;

                highestPricePid = products[i].pid;

            }

        }

        System.out.println("Product ID with the highest price: " + highestPricePid);

        // Call method to calculate total amount spent on products

        calculateTotalAmount(products);

    }

}

**OUTPUT**

A screenshot of a computer

AI-generated content may be incorrect.

**3.Create Class Account with data member as Balance . Create two constructors and  perform following task**

1. **method to deposit the amount to the account .**
2. **method to withdraw the amount from the account .**
3. **method to display the Balance  .**

**CODE**

// Define a class named Account

public class Account {

    // variable to store the account balance

    double balance;

    // Default constructor

    public Account() {

        System.out.println("Bank account");

    }

    // Parameterized constructor to initialize balance

    public Account(double balance) {

        this.balance = balance;

    }

    // Method to deposit money into the account

    void deposit(double amount) {

        balance = amount + balance;

        System.out.println("Deposit amount - " + amount);

    }

    // Method to withdraw money from the account

    void withdraw(double amount) {

        if (amount <= balance) {

            balance = balance - amount;

            System.out.println("Withdraw amount - " + amount);

        } else {

            System.out.println("insufficient balance");

        }

    }

    // Method to display the account balance

    void displaybalance() {

        System.out.println("Balance - " + balance);

    }

    public static void main(String[] args) {

        // Create an Account object

        Account p1 = new Account(15000);

        // Deposit 5000 into the account

        p1.deposit(5000);

        // Display the updated balance

        p1.displaybalance();

        // Withdraw 3000 from the account

        p1.withdraw(3000);

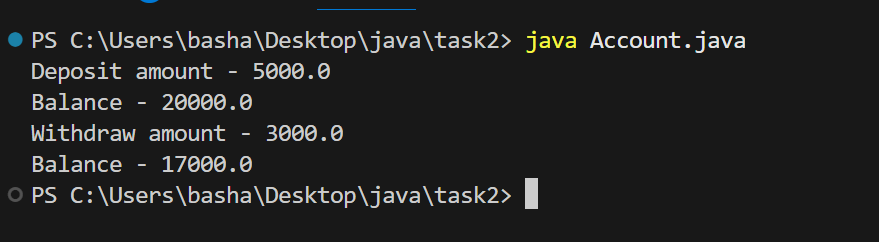
        // Display the final balance

        p1.displaybalance();

    }

}

**OUTPUT**

****

**4.Define a base class Person with attributes 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.**

**CODE**

// Define a class named Person

public class Person {

    // Declare  variables

    String name;

    int age;

    // Constructor to initialize name and age

    Person(String name, int age) {

        this.name = name;

        this.age = age;

    }

    public static void main(String[] args) {

        // Create an Employee object

        Employee e1 = new Employee("basha", 20);

        // Print the Employee's name and age

        System.out.println(e1.name);

        System.out.println(e1.age);

    }

}

// Define a subclass Employee that extends Person

class Employee extends Person {

    // Constructor for Employee

    Employee(String name, int age) {

        super(name, age); // Calls the constructor of the Person class using super keyword

        System.out.println("Employee constructor");

    }

}

**OUTPUT**

A screen shot of a computer

AI-generated content may be incorrect.