Constructors, Java Keywords - Assignment Problem (Any 4)

# Problem 1: Movie Ticket Booking 🎬

Design a **Movie Ticket System**.

* Class MovieTicket with fields: String movieName, String theatreName, int seatNumber, double price.
* Implement constructor overloading:
  1. Default constructor → assigns "Unknown" movie.
  2. Constructor with movie name → assigns default price = 200.
  3. Constructor with movie name and seat number → assigns default theatre "PVR".
  4. Full constructor → sets all details.
* Add method: printTicket() → displays ticket details.
* In main(): Create and print multiple tickets.

public class MovieTicket {

String movieName, theatreName;

int seatNumber;

double price;

MovieTicket() {

movieName = "Unknown";

}

MovieTicket(String movieName) {

this.movieName = movieName;

price = 200;

}

MovieTicket(String movieName, int seatNumber) {

this.movieName = movieName;

this.seatNumber = seatNumber;

theatreName = "PVR";

}

MovieTicket(String movieName, String theatreName, int seatNumber, double price) {

this.movieName = movieName;

this.theatreName = theatreName;

this.seatNumber = seatNumber;

this.price = price;

}

void printTicket() {

System.out.println("Movie: " + movieName + ", Theatre: " + theatreName + ", Seat: " + seatNumber + ", Price: " + price);

}

public static void main(String[] args) {

MovieTicket t1 = new MovieTicket();

MovieTicket t2 = new MovieTicket("Avengers");

MovieTicket t3 = new MovieTicket("Inception", 15);

MovieTicket t4 = new MovieTicket("Interstellar", "Cineplex", 22, 350);

t1.printTicket();

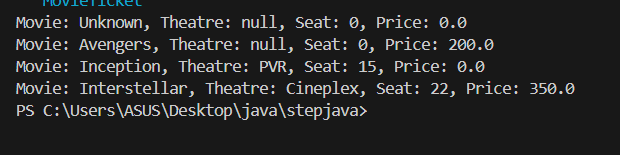
t2.printTicket();

t3.printTicket();

t4.printTicket();

}

}



# Problem 2: Bank Account System 💳

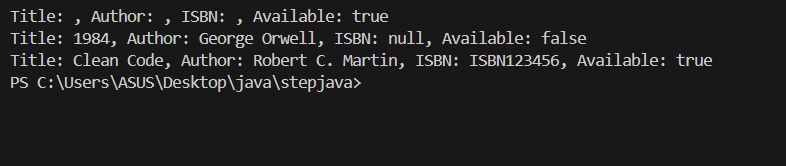
Create a **Bank Account** management program.

* Class BankAccount with fields: String accountHolder, int accountNumber, double balance.
* Implement constructor overloading:
* Default constructor → balance = 0.
* Constructor with name → assigns random account number.
* Constructor with name and initial balance → assigns both.
* Add methods:
* deposit(double amount)
* withdraw(double amount)
* displayAccount()
* In main(): Create accounts, deposit/withdraw, and display balance.

# Problem 3: Library Book Management 📚

Design a system for managing **Library Books**.

* Class Book with fields: String title, String author, String isbn, boolean isAvailable.
* Constructor overloading:
* Default constructor → empty book.
* Constructor with title and author.
* Constructor with all details.
* Methods:
* borrowBook() → sets available = false.
* returnBook() → sets available = true.
* displayBookInfo().
* In main(): Create books, borrow/return them, display info.
* public class Book3 {
* String title, author, isbn;
* boolean isAvailable;
* Book3() {
* title = "";
* author = "";
* isbn = "";
* isAvailable = true;
* }
* Book3(String title, String author) {
* this.title = title;
* this.author = author;
* isAvailable = true;
* }
* Book3(String title, String author, String isbn, boolean isAvailable) {
* this.title = title;
* this.author = author;
* this.isbn = isbn;
* this.isAvailable = isAvailable;
* }
* void borrowBook() {
* isAvailable = false;
* }
* void returnBook() {
* isAvailable = true;
* }
* void displayBookInfo() {
* System.out.println("Title: " + title + ", Author: " + author + ", ISBN: " + isbn + ", Available: " + isAvailable);
* }
* public static void main(String[] args) {
* Book3 b1 = new Book3();
* Book3 b2 = new Book3("1984", "George Orwell");
* Book3 b3 = new Book3("Clean Code", "Robert C. Martin", "ISBN123456", true);
* b2.borrowBook();
* b3.returnBook();
* b1.displayBookInfo();
* b2.displayBookInfo();
* b3.displayBookInfo();
* }
* }



# Problem 4: Food Delivery Order 🍔

Create a program to simulate a **Food Delivery System**.

* Class FoodOrder with fields: String customerName, String foodItem, int quantity, double price.
* Constructor overloading:
  1. Default constructor → assigns "Unknown" order.
  2. Constructor with food item → sets quantity = 1, price = default.
  3. Constructor with food item and quantity → calculates price = quantity × fixedRate.
* Method: printBill() → displays order details and total price.
* In main(): Create multiple orders and print bills.

public class FoodOrder {

String customerName, foodItem;

int quantity;

double price;

FoodOrder() {

customerName = "Unknown";

foodItem = "Unknown";

quantity = 0;

price = 0;

}

FoodOrder(String foodItem) {

this.foodItem = foodItem;

quantity = 1;

price = 100;

}

FoodOrder(String foodItem, int quantity) {

this.foodItem = foodItem;

this.quantity = quantity;

price = quantity \* 100;

}

void printBill() {

System.out.println("Customer: " + customerName + ", Food: " + foodItem + ", Quantity: " + quantity + ", Total Price: " + price);

}

public static void main(String[] args) {

FoodOrder o1 = new FoodOrder();

FoodOrder o2 = new FoodOrder("Burger");

FoodOrder o3 = new FoodOrder("Pizza", 3);

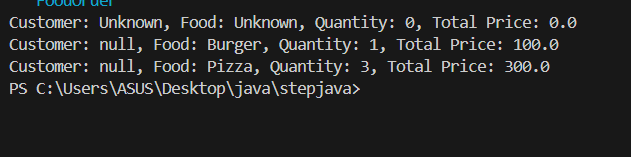
o1.printBill();

o2.printBill();

o3.printBill();

}

}



# Problem 5: Fitness Tracker 🏃

Design a **Fitness Tracker App** simulation.

* Class Workout with fields: String activityName, int durationInMinutes, int caloriesBurned.
* Constructor overloading:
  1. Default constructor → "Walking", 30 mins, 100 calories.
  2. Constructor with activity name → assigns default duration.
  3. Constructor with activity and duration → calculate caloriesBurned = duration × 5.
* Method: displayWorkout() → prints activity details.
* In main(): Create different workouts and display details.

public class Workout {

    String activityName;

    int durationInMinutes;

    int caloriesBurned;

    Workout() {

        activityName = "Walking";

        durationInMinutes = 30;

        caloriesBurned = 100;

    }

    Workout(String activityName) {

        this.activityName = activityName;

        durationInMinutes = 30;

        caloriesBurned = durationInMinutes \* 5;

    }

    Workout(String activityName, int durationInMinutes) {

        this.activityName = activityName;

        this.durationInMinutes = durationInMinutes;

        caloriesBurned = durationInMinutes \* 5;

    }

    void displayWorkout() {

        System.out.println("Activity: " + activityName + ", Duration: " + durationInMinutes + " mins, Calories Burned: " + caloriesBurned);

    }

    public static void main(String[] args) {

        Workout w1 = new Workout();

        Workout w2 = new Workout("Running");

        Workout w3 = new Workout("Cycling", 45);

        w1.displayWorkout();

        w2.displayWorkout();

        w3.displayWorkout();

    }

}

