```java
//  Food Delivery Order
// 
// 

// Create a program to simulate a Food Delivery System.
// 2
// • 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.


class FoodOrder {
    private String customerName;
    private String foodItem;
    private int quantity;
    private double price;

    private static final double FIXED_RATE = 150.0;

    // Default constructor
    public FoodOrder() {
        this("Unknown", "Unknown", 0, 0.0);
    }

    // Constructor with food item only
    public FoodOrder(String foodItem) {
        this("Customer", foodItem, 1, FIXED_RATE);
    }

    // Constructor with food item and quantity
    public FoodOrder(String foodItem, int quantity) {
        this("Customer", foodItem, quantity, quantity * FIXED_RATE);
```

```
    }

    // Full constructor
    public FoodOrder(String customerName, String foodItem, int quantity,
double price) {
        this.customerName = customerName;
        this.foodItem = foodItem;
        this.quantity = quantity;
        this.price = price;
    }

    // Method to print bill
    public void printBill() {
        System.out.println("🧾 Food Order Bill");
        System.out.println("Customer: " + customerName);
        System.out.println("Food Item: " + foodItem);
        System.out.println("Quantity: " + quantity);
        System.out.println("Total Price: ₹" + price);
        System.out.println("----------------------------");
    }
}

public class FoodDeliverySystem {
    public static void main(String[] args) {
        FoodOrder order1 = new FoodOrder();
        FoodOrder order2 = new FoodOrder("Burger");
        FoodOrder order3 = new FoodOrder("Pizza", 3);
        FoodOrder order4 = new FoodOrder("Aryan", "Pasta", 2, 300.0);

        order1.printBill();
        order2.printBill();
        order3.printBill();
        order4.printBill();
    }
}
```

```
PS E:\JAVA PROGRAMS\steparyansingh\year2\oops\week4\assignment> run FoodDeliverySystem
Compiling FoodDeliverySystem.java...
Compilation successful. Running program...

? Food Order Bill
Customer: Unknown
Food Item: Unknown
Quantity: 0
Total Price: ?0.0
--------------------------
? Food Order Bill
Customer: Customer
Food Item: Burger
Quantity: 1
Total Price: ?150.0
--------------------------
? Food Order Bill
Customer: Customer
Food Item: Pizza
Quantity: 3
Total Price: ?450.0
--------------------------
? Food Order Bill
Customer: Aryan
Food Item: Pasta
Quantity: 2
Total Price: ?300.0
--------------------------

Program finished. Cleaning up...
FoodDeliverySystem.class file deleted successfully.
Press any key to continue . . .
```

```
//  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:
// 1
// ○ deposit(double amount)
// ○ withdraw(double amount)
```

```java
// ○ displayAccount()
// ● In main(): Create accounts, deposit/withdraw, and display balance.


import java.util.*;

class BankAccount {
    private String accountHolder;
    private int accountNumber;
    private double balance;

    // Default constructor
    public BankAccount() {
        this("Unknown", 0.0);
    }

    // Constructor with name only
    public BankAccount(String accountHolder) {
        this(accountHolder, 0.0);
    }

    // Constructor with name and initial balance
    public BankAccount(String accountHolder, double balance) {
        this.accountHolder = accountHolder;
        this.accountNumber = generateAccountNumber();
        this.balance = balance;
    }

    // Random account number generator
    private int generateAccountNumber() {
        return new Random().nextInt(900000) + 100000; // 6-digit number
    }

    // Deposit method
    public void deposit(double amount) {
        if (amount > 0) {
            balance += amount;
            System.out.println("Deposited ₹" + amount + " to " +
accountHolder + "'s account.");
        } else {
```

```
            System.out.println("Invalid deposit amount.");
        }
    }

    // Withdraw method
    public void withdraw(double amount) {
        if (amount > 0 && amount <= balance) {
            balance -= amount;
            System.out.println("Withdrew ₹" + amount + " from " +
accountHolder + "'s account.");
        } else {
            System.out.println("Insufficient balance or invalid amount.");
        }
    }

    // Display account details
    public void displayAccount() {
        System.out.println("🏦 Account Holder: " + accountHolder);
        System.out.println("Account Number: " + accountNumber);
        System.out.println("Balance: ₹" + balance);
        System.out.println("----------------------------");
    }
}

public class BankAccountSystem {
    public static void main(String[] args) {
        BankAccount acc1 = new BankAccount();
        BankAccount acc2 = new BankAccount("Aryan");
        BankAccount acc3 = new BankAccount("Riya", 5000.0);

        acc1.deposit(1000);
        acc2.deposit(2000);
        acc3.withdraw(1500);

        acc1.displayAccount();
        acc2.displayAccount();
        acc3.displayAccount();
    }
}
```

```
PS E:\JAVA PROGRAMS\steparyansingh\year2\oops\week4\assignment> run BankAccountSystem
Compiling BankAccountSystem.java...
Compilation successful. Running program...

Deposited ?1000.0 to Unknown's account.
Deposited ?2000.0 to Aryan's account.
Withdrew ?1500.0 from Riya's account.
? Account Holder: Unknown
Account Number: 827851
Balance: ?1000.0
--------------------------
? Account Holder: Aryan
Account Number: 525528
Balance: ?2000.0
--------------------------
? Account Holder: Riya
Account Number: 747164
Balance: ?3500.0
--------------------------

Program finished. Cleaning up...
BankAccountSystem.class file deleted successfully.
Press any key to continue . . .
PS E:\JAVA PROGRAMS\steparyansingh\year2\oops\week4\assignment>
```

```java
//  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.


class Book {
    private String title;
    private String author;
```

```
    private String isbn;
    private boolean isAvailable;

    // Default constructor
    public Book() {
        this("Untitled", "Unknown", "0000000000", true);
    }

    // Constructor with title and author
    public Book(String title, String author) {
        this(title, author, "0000000000", true);
    }

    // Full constructor
    public Book(String title, String author, String isbn, boolean
isAvailable) {
        this.title = title;
        this.author = author;
        this.isbn = isbn;
        this.isAvailable = isAvailable;
    }

    // Borrow the book
    public void borrowBook() {
        if (isAvailable) {
            isAvailable = false;
            System.out.println("📕 '" + title + "' has been borrowed.");
        } else {
            System.out.println("⚠️ '" + title + "' is already borrowed.");
        }
    }

    // Return the book
    public void returnBook() {
        if (!isAvailable) {
            isAvailable = true;
            System.out.println("📗 '" + title + "' has been returned.");
        } else {
            System.out.println("ℹ️ '" + title + "' was not borrowed.");
        }
```

```
    }

    // Display book info
    public void displayBookInfo() {
        System.out.println("📖 Title: " + title);
        System.out.println("Author: " + author);
        System.out.println("ISBN: " + isbn);
        System.out.println("Available: " + (isAvailable ? "Yes" : "No"));
        System.out.println("----------------------------");
    }
}

public class LibraryBookManagement {
    public static void main(String[] args) {
        Book b1 = new Book();
        Book b2 = new Book("The Alchemist", "Paulo Coelho");
        Book b3 = new Book("1984", "George Orwell", "9780451524935",
true);

        b1.displayBookInfo();
        b2.displayBookInfo();
        b3.displayBookInfo();

        b2.borrowBook();
        b2.displayBookInfo();

        b2.returnBook();
        b2.displayBookInfo();
    }
}
```

```
PS E:\JAVA PROGRAMS\steparyansingh\year2\oops\week4\assignment> run LibraryBookManagement
Compiling LibraryBookManagement.java...
Compilation successful. Running program...

? Title: Untitled
Author: Unknown
ISBN: 0000000000
Available: Yes
--------------------------
? Title: The Alchemist
Author: Paulo Coelho
ISBN: 0000000000
Available: Yes
--------------------------
? Title: 1984
Author: George Orwell
ISBN: 9780451524935
Available: Yes
--------------------------
? 'The Alchemist' has been borrowed.
? Title: The Alchemist
Author: Paulo Coelho
ISBN: 0000000000
Available: No
--------------------------
? 'The Alchemist' has been returned.
? Title: The Alchemist
Author: Paulo Coelho
ISBN: 0000000000
Available: Yes
--------------------------

Program finished. Cleaning up...
LibraryBookManagement.class file deleted successfully.
Press any key to continue . . .
```

```java
class MovieTicket {
    private String movieName;
    private String theatreName;
    private int seatNumber;
    private double price;

    // Default constructor
    public MovieTicket() {
        this("Unknown", "Generic Theatre", -1, 0.0);
    }

    // Constructor with movie name
    public MovieTicket(String movieName) {
        this(movieName, "Generic Theatre", -1, 200.0);
    }

    // Constructor with movie name and seat number
    public MovieTicket(String movieName, int seatNumber) {
```

```
        this(movieName, "PVR", seatNumber, 200.0);
    }

    // Full constructor
    public MovieTicket(String movieName, String theatreName, int
seatNumber, double price) {
        this.movieName = movieName;
        this.theatreName = theatreName;
        this.seatNumber = seatNumber;
        this.price = price;
    }

    // Method to print ticket details
    public void printTicket() {
        System.out.println("🎟️ Movie Ticket");
        System.out.println("Movie: " + movieName);
        System.out.println("Theatre: " + theatreName);
        System.out.println("Seat No: " + seatNumber);
        System.out.println("Price: ₹" + price);
        System.out.println("-------------------------");
    }
}

public class MovieTicketBookingSystem {
    public static void main(String[] args) {
        MovieTicket t1 = new MovieTicket();
        MovieTicket t2 = new MovieTicket("Inception");
        MovieTicket t3 = new MovieTicket("Interstellar", 12);
        MovieTicket t4 = new MovieTicket("Tenet", "IMAX", 5, 350.0);

        t1.printTicket();
        t2.printTicket();
        t3.printTicket();
        t4.printTicket();
    }
}
```

```
PS E:\JAVA PROGRAMS\steparyansingh\year2\oops\week4\assignment> run MovieTicketBookingSystem
Compiling MovieTicketBookingSystem.java...
Compilation successful. Running program...

?? Movie Ticket
Movie: Unknown
Theatre: Generic Theatre
Seat No: -1
Price: ?0.0
------------------------
?? Movie Ticket
Movie: Inception
Theatre: Generic Theatre
Seat No: -1
Price: ?200.0
------------------------
?? Movie Ticket
Movie: Interstellar
Theatre: PVR
Seat No: 12
Price: ?200.0
------------------------
?? Movie Ticket
Movie: Tenet
Theatre: IMAX
Seat No: 5
Price: ?350.0
------------------------

Program finished. Cleaning up...
MovieTicketBookingSystem.class file deleted successfully.
Press any key to continue . . .
```