# FAZAIA BILQUIS COLLEGE
## NUR KHAN BASE,RWP

## Lab #02 Classes & Objects

SUBMITTED BY: *MARIA ATTA*

ROLL NO: *BSCS-13-F24-01*

SUBJECT: *OOP'S*

SUBMITTED TO: *MS.SANA*

# LAB 2: CLASSES AND OBJECTS IN JAVA

## 1. INTRODUCTION TO CLASSES AND OBJECTS IN JAVA

- ### WHAT IS A CLASS?

A class is like a blueprint. It defines what an object will look like and what it can do. For example:

- Think of a class as a blueprint for a house. The blueprint defines the structure of the house (e.g., number of rooms, doors, windows).
- The object is the actual house built using that blueprint.

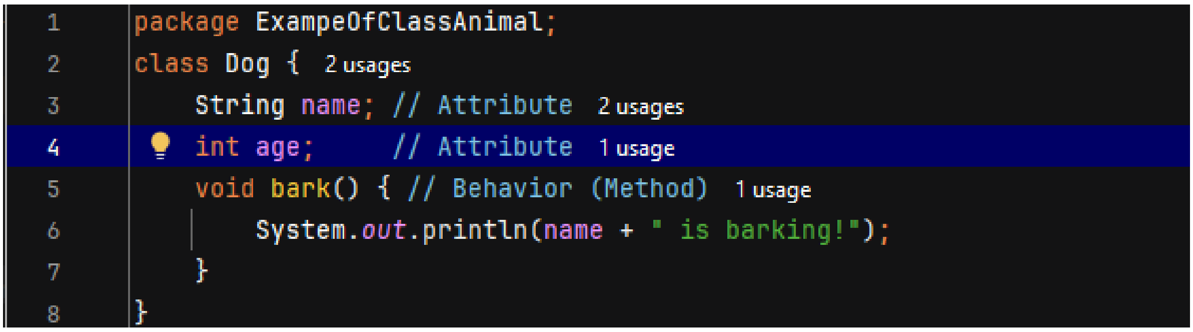
- ### WHAT IS AN OBJECT?

An object is an instance of a class. It is a real-world entity that has:

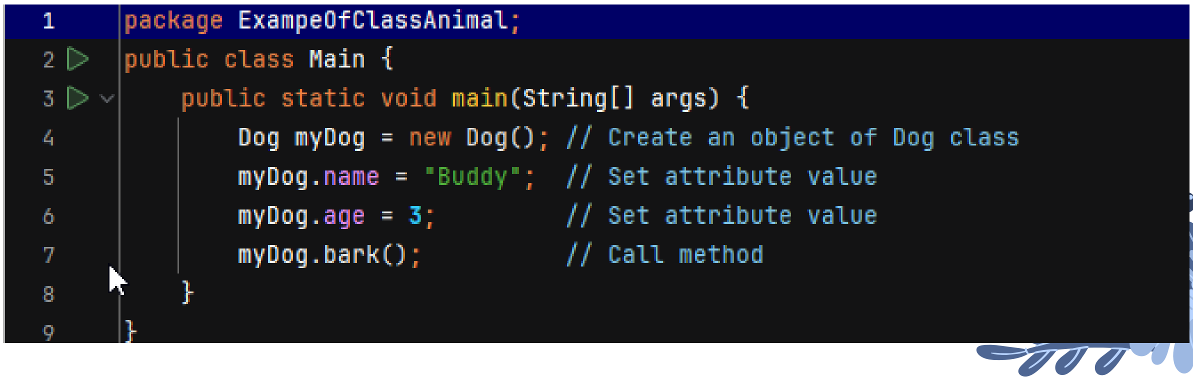
- Attributes: Properties or data (e.g., name, age, color).
- Behaviors: Actions or methods (e.g., run, eat, display).

**_EXAMPLE OF CLASS AND OBJECT:_**

- ***Program:***

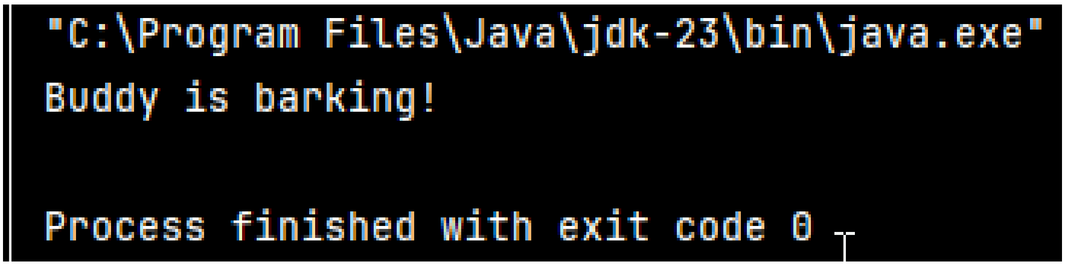


```
package ExampeOfClassAnimal;
class Dog {  2 usages
    String name; // Attribute  2 usages
    int age;     // Attribute  1 usage
    void bark() { // Behavior (Method)  1 usage
        System.out.println(name + " is barking!");
    }
}
```



```
package ExampeOfClassAnimal;
public class Main {
    public static void main(String[] args) {
        Dog myDog = new Dog(); // Create an object of Dog class
        myDog.name = "Buddy";  // Set attribute value
        myDog.age = 3;         // Set attribute value
        myDog.bark();          // Call method
    }
}
```

- ***Output:***



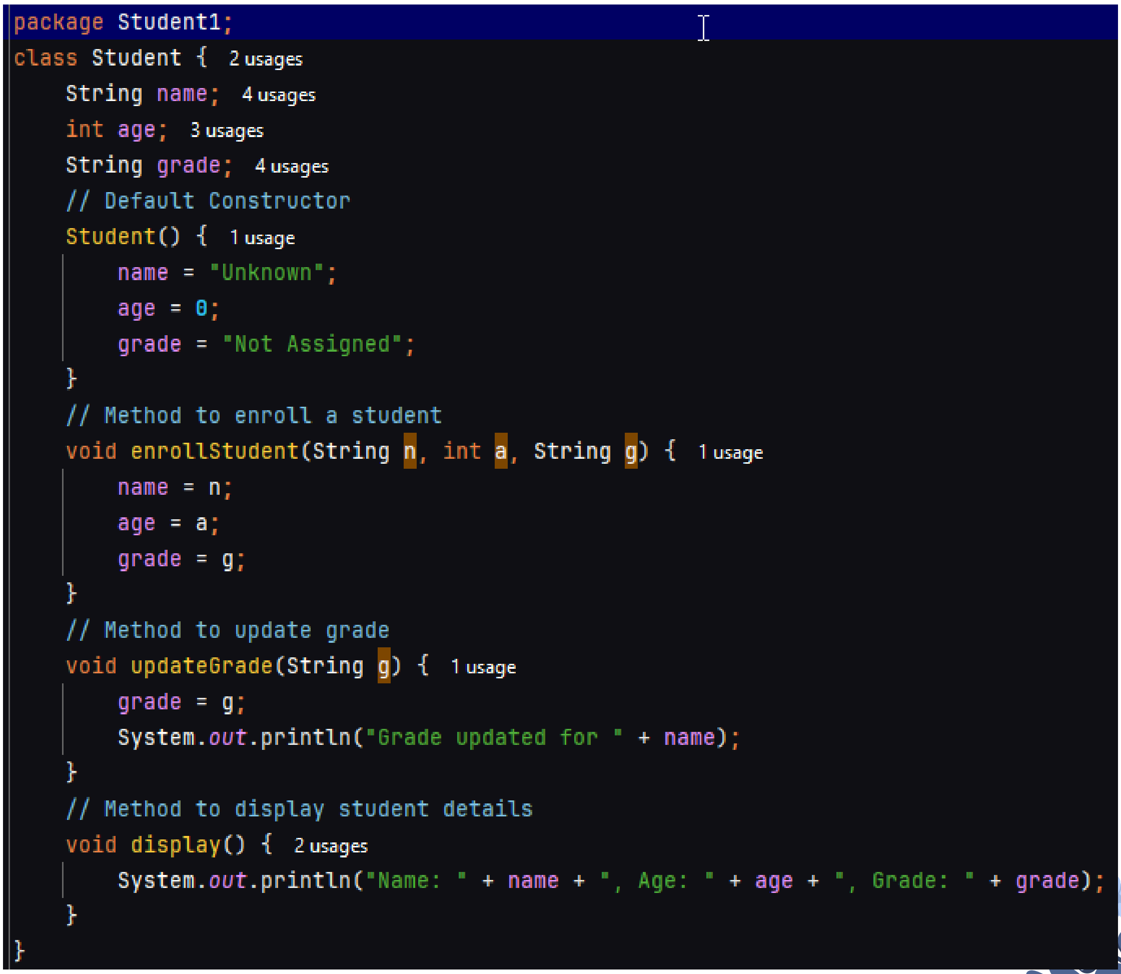
```
"C:\Program Files\Java\jdk-23\bin\java.exe"
Buddy is barking!

Process finished with exit code 0
```

## PROGRAM 1: STUDENT MANAGEMENT SYSTEM:

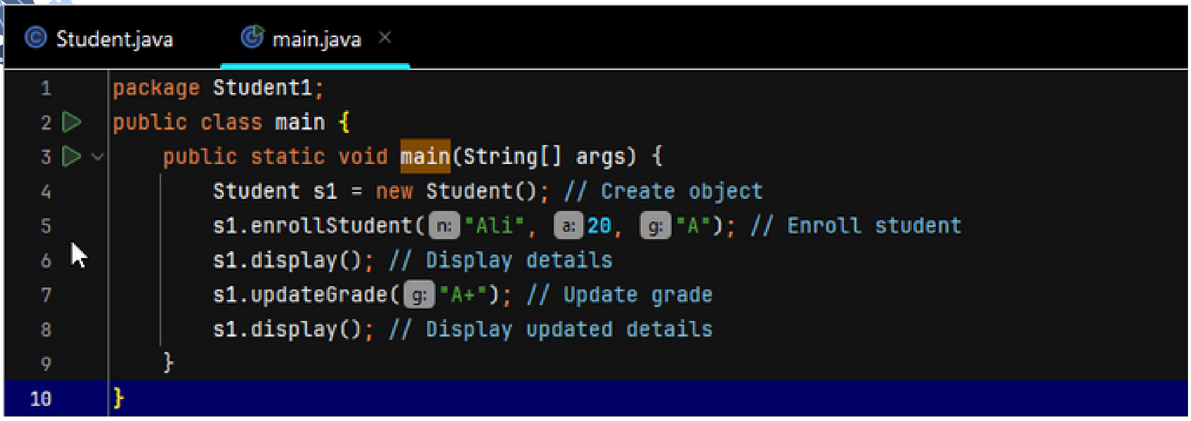
**Problem:** Create a student class with attributes such as name, age, and grade. Add methods to:

1. Enroll a student by setting their details.
2. Update student details.
3. Display student information.

- ***Program:***

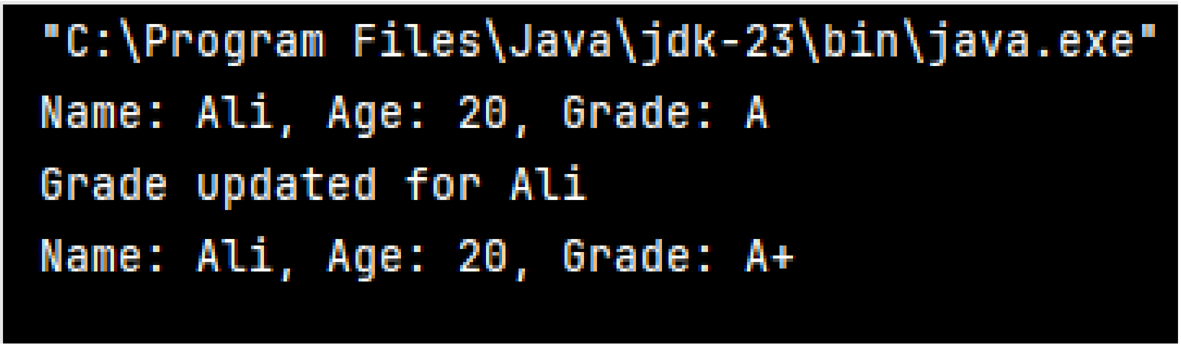


```
package Student1;
class Student {  2 usages
    String name;  4 usages
    int age;  3 usages
    String grade;  4 usages
    // Default Constructor
    Student() {  1 usage
        name = "Unknown";
        age = 0;
        grade = "Not Assigned";
    }
    // Method to enroll a student
    void enrollStudent(String n, int a, String g) {  1 usage
        name = n;
        age = a;
        grade = g;
    }
    // Method to update grade
    void updateGrade(String g) {  1 usage
        grade = g;
        System.out.println("Grade updated for " + name);
    }
    // Method to display student details
    void display() {  2 usages
        System.out.println("Name: " + name + ", Age: " + age + ", Grade: " + grade);
    }
}
```



```java
package Student1;
public class main {
    public static void main(String[] args) {
        Student s1 = new Student(); // Create object
        s1.enrollStudent("Ali", 20, "A"); // Enroll student
        s1.display(); // Display details
        s1.updateGrade("A+"); // Update grade
        s1.display(); // Display updated details
    }
}
```

- ***Output:***



```
"C:\Program Files\Java\jdk-23\bin\java.exe"
Name: Ali, Age: 20, Grade: A
Grade updated for Ali
Name: Ali, Age: 20, Grade: A+
```

- ***Code Explanation:***

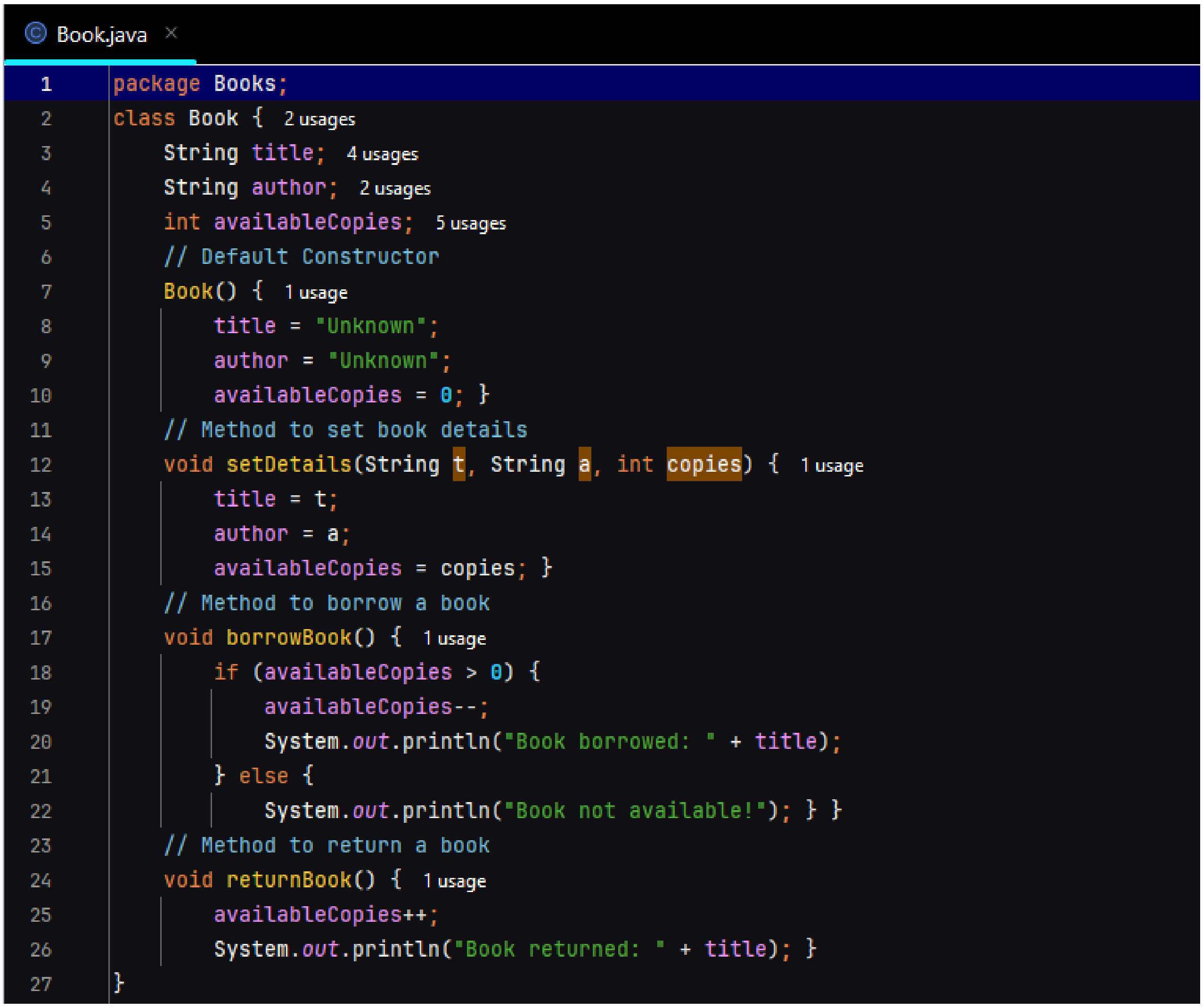
1. **Class:** Student has attributes (name, age, grade) and methods (enroll student, updateGrade, display).
2. **Object:** **s1** is an object of the Student class.
3. **Methods:**
    - **enrollStudent:** Sets student details.
    - **updateGrade:** Updates the grade.
    - **display:** Shows student details.

## PROGRAM 2: LIBRARY BOOK SYSTEM

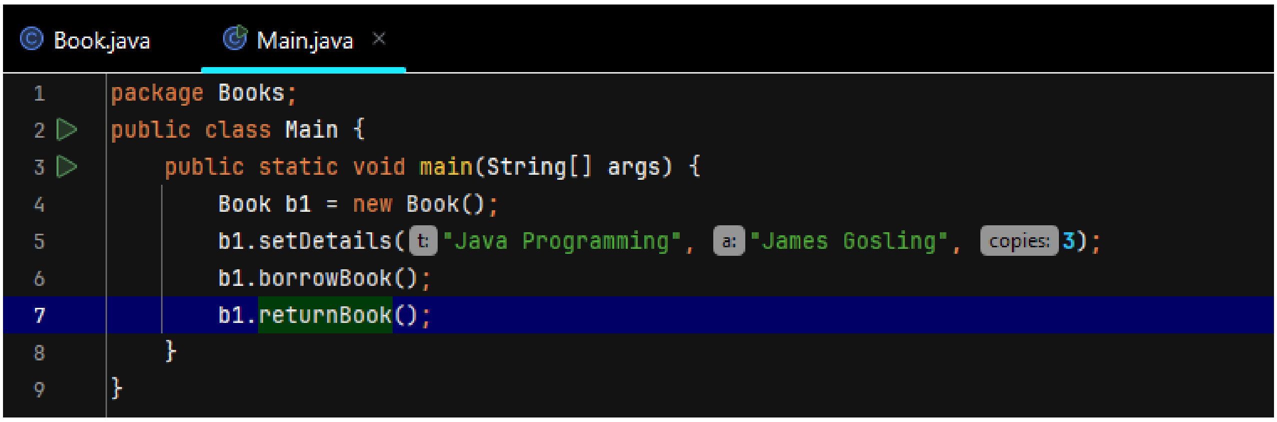
**Problem:** Create a book class with attributes such as title, author, and available copies. Add methods to:

1. Set book details.
2. Borrow a book (reduce available copies).
3. Return a book (increase available copies).

## • Program:

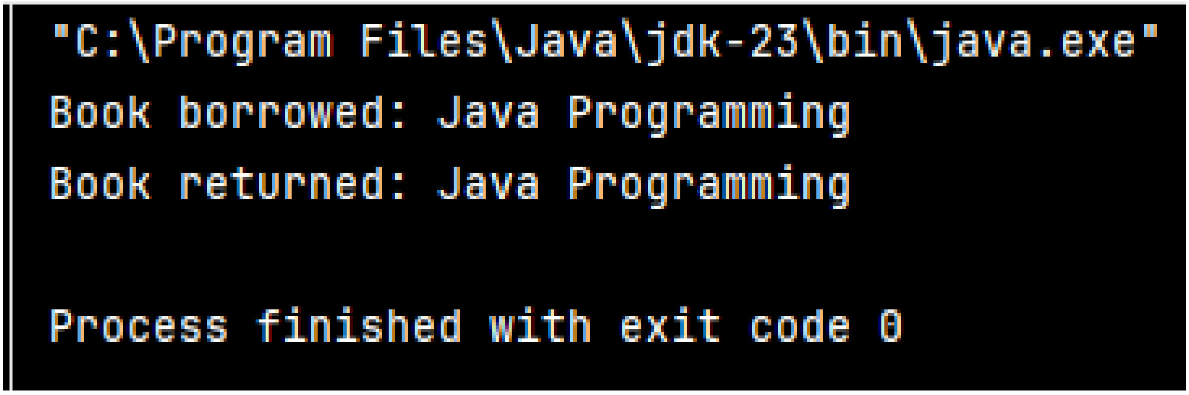


```java
package Books;
class Book {  2 usages
    String title;  4 usages
    String author;  2 usages
    int availableCopies;  5 usages
    // Default Constructor
    Book() {  1 usage
        title = "Unknown";
        author = "Unknown";
        availableCopies = 0; }
    // Method to set book details
    void setDetails(String t, String a, int copies) {  1 usage
        title = t;
        author = a;
        availableCopies = copies; }
    // Method to borrow a book
    void borrowBook() {  1 usage
        if (availableCopies > 0) {
            availableCopies--;
            System.out.println("Book borrowed: " + title);
        } else {
            System.out.println("Book not available!"); } }
    // Method to return a book
    void returnBook() {  1 usage
        availableCopies++;
        System.out.println("Book returned: " + title); }
}
```



```java
package Books;
public class Main {
    public static void main(String[] args) {
        Book b1 = new Book();
        b1.setDetails(t: "Java Programming", a: "James Gosling", copies: 3);
        b1.borrowBook();
        b1.returnBook();
    }
}
```

## • Output:



```
"C:\Program Files\Java\jdk-23\bin\java.exe"
Book borrowed: Java Programming
Book returned: Java Programming

Process finished with exit code 0
```

- ***Code Explanation:***

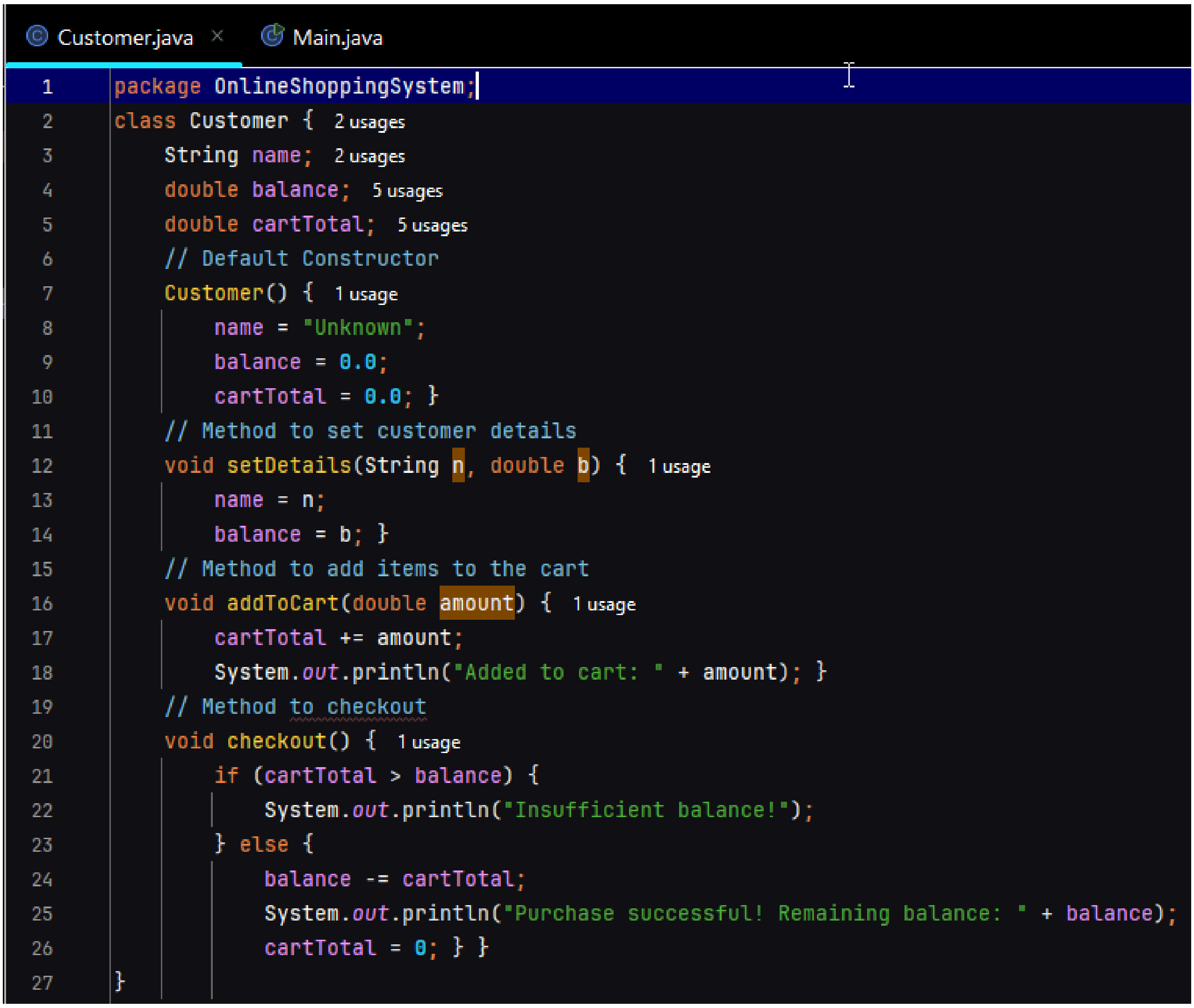
1. **Class:** Book has attributes (title, author, available) and methods (setDetails, borrowBook, returnBook).
2. **Object:** b1 is an object of the Book class.
3. **Methods:**
    - **setDetails:** Sets book details.
    - **borrowBook:** Reduces available copies.
    - **returnBook:** Increases available copies.

## PROGRAM 3: ONLINE SHOPPING SYSTEM

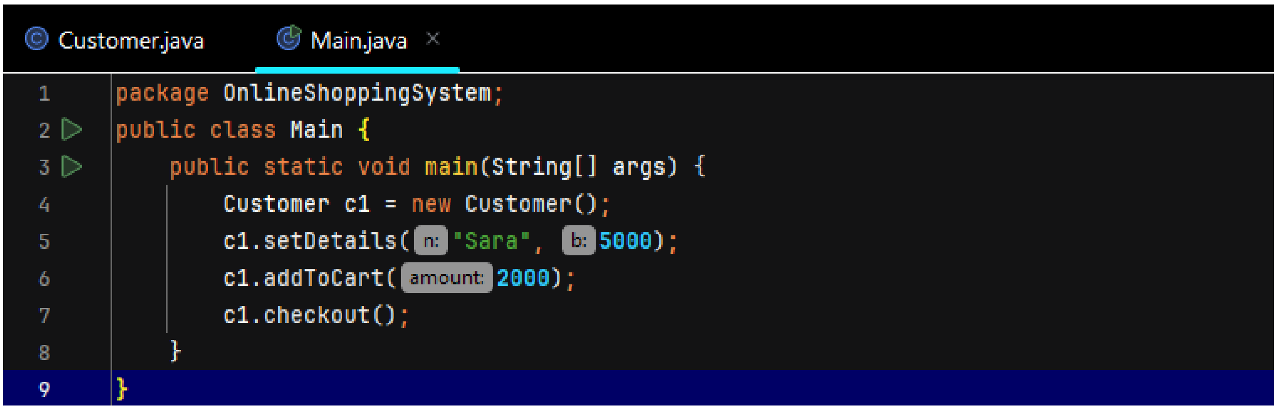
**Problem:** Create a Customer class with attributes name, balance, and cartTotal. Add methods to:

1. Add items to the cart.
2. Checkout (reduce balance if sufficient funds are available).

- ***Program:***

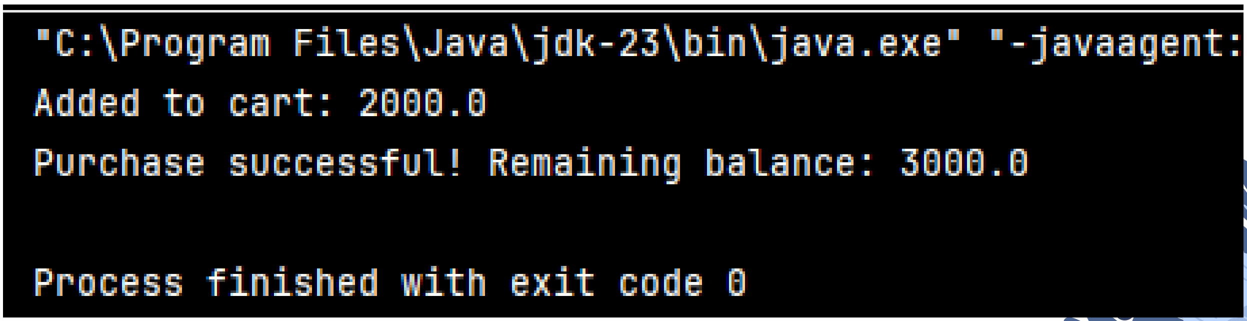


```
package OnlineShoppingSystem;
class Customer {  2 usages
    String name;  2 usages
    double balance;  5 usages
    double cartTotal;  5 usages
    // Default Constructor
    Customer() {  1 usage
        name = "Unknown";
        balance = 0.0;
        cartTotal = 0.0; }
    // Method to set customer details
    void setDetails(String n, double b) {  1 usage
        name = n;
        balance = b; }
    // Method to add items to the cart
    void addToCart(double amount) {  1 usage
        cartTotal += amount;
        System.out.println("Added to cart: " + amount); }
    // Method to checkout
    void checkout() {  1 usage
        if (cartTotal > balance) {
            System.out.println("Insufficient balance!");
        } else {
            balance -= cartTotal;
            System.out.println("Purchase successful! Remaining balance: " + balance);
            cartTotal = 0; } }
}
```



```
package OnlineShoppingSystem;
public class Main {
    public static void main(String[] args) {
        Customer c1 = new Customer();
        c1.setDetails( n: "Sara",  b: 5000);
        c1.addToCart( amount: 2000);
        c1.checkout();
    }
}
```

- ***Output:***



```
"C:\Program Files\Java\jdk-23\bin\java.exe" "-javaagent:
Added to cart: 2000.0
Purchase successful! Remaining balance: 3000.0

Process finished with exit code 0
```

- ## *Code Explanation:*

1. **Class:** Customer has attributes (name, balance, cartTotal) and methods (setDetails, addToCart, checkout).
2. **Object:** c1 is an object of the Customer class.
3. **Methods:**
    - **setDetails:** Sets customer details.
    - **addToCart:** Adds items to the cart.
    - **checkout:** Deducts the cart total from the balance.

- ## *Conclusion:*

1. **Classes** are blueprints for creating objects.
2. **Objects** are instances of classes with attributes and behaviors.
3. **Constructors** initialize objects with default values.
4. **Methods** define the actions an object can perform.

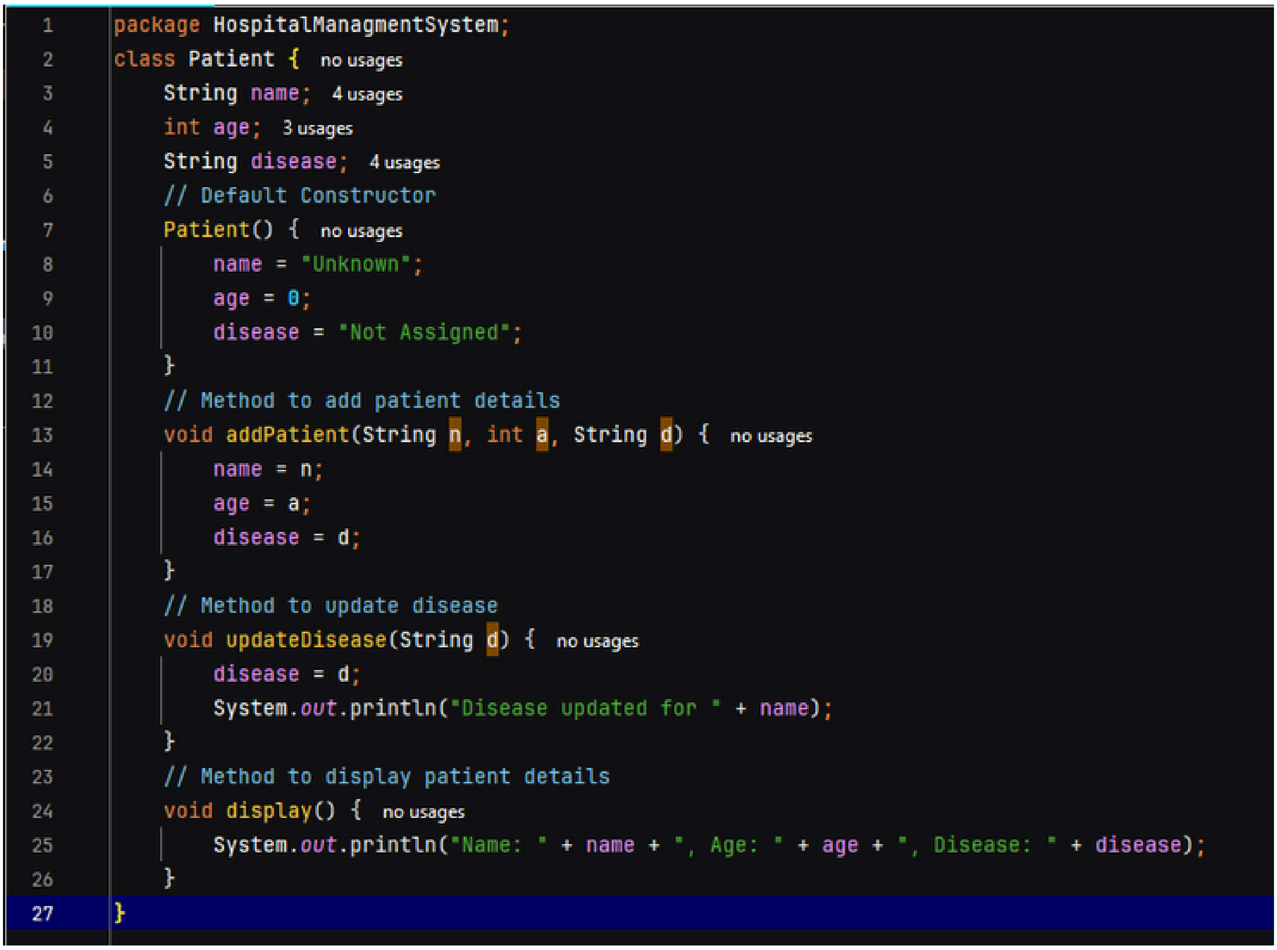
## HOME TASKS:

### Q1. HOSPITAL PATIENT MANAGEMENT SYSTEM

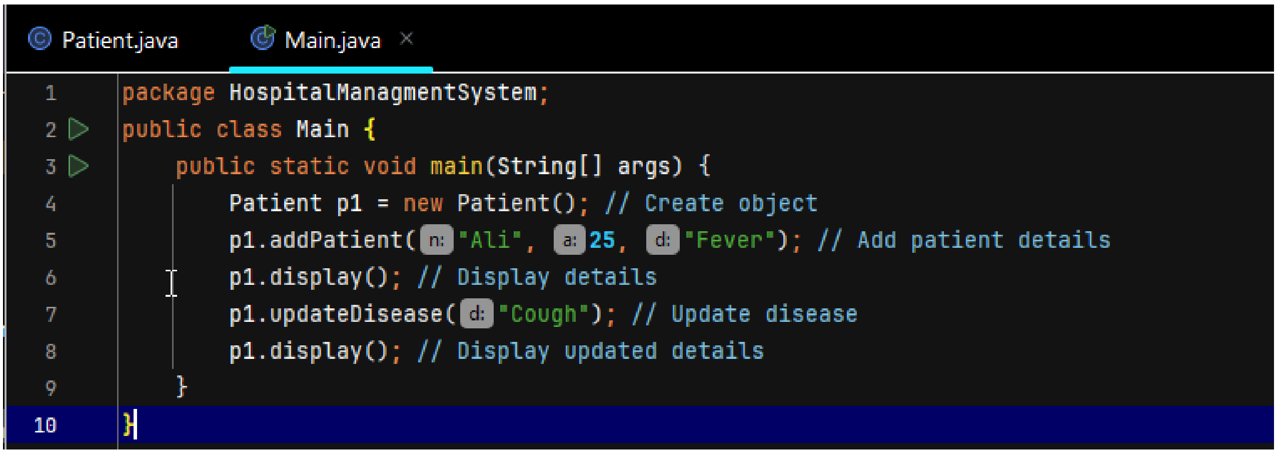
**Problem:** Create a Patient class to manage patient details. The class should allow:

1. Adding a new patient with details like name, age, and disease.
2. Updating a patient's disease status.
3. Displaying patient details.

- ## *Program:*

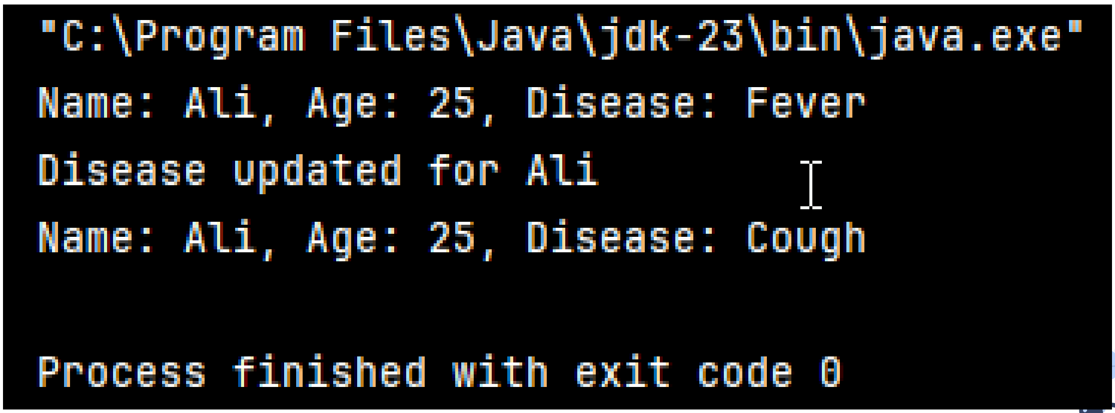


```java
package HospitalManagmentSystem;
class Patient {  no usages
    String name;  4 usages
    int age;  3 usages
    String disease;  4 usages
    // Default Constructor
    Patient() {  no usages
        name = "Unknown";
        age = 0;
        disease = "Not Assigned";
    }
    // Method to add patient details
    void addPatient(String n, int a, String d) {  no usages
        name = n;
        age = a;
        disease = d;
    }
    // Method to update disease
    void updateDisease(String d) {  no usages
        disease = d;
        System.out.println("Disease updated for " + name);
    }
    // Method to display patient details
    void display() {  no usages
        System.out.println("Name: " + name + ", Age: " + age + ", Disease: " + disease);
    }
}
```



```java
package HospitalManagmentSystem;
public class Main {
    public static void main(String[] args) {
        Patient p1 = new Patient(); // Create object
        p1.addPatient( n: "Ali",  a: 25,  d: "Fever"); // Add patient details
        p1.display(); // Display details
        p1.updateDisease( d: "Cough"); // Update disease
        p1.display(); // Display updated details
    }
}
```

- ***Output:***



```
"C:\Program Files\Java\jdk-23\bin\java.exe"
Name: Ali, Age: 25, Disease: Fever
Disease updated for Ali
Name: Ali, Age: 25, Disease: Cough

Process finished with exit code 0
```

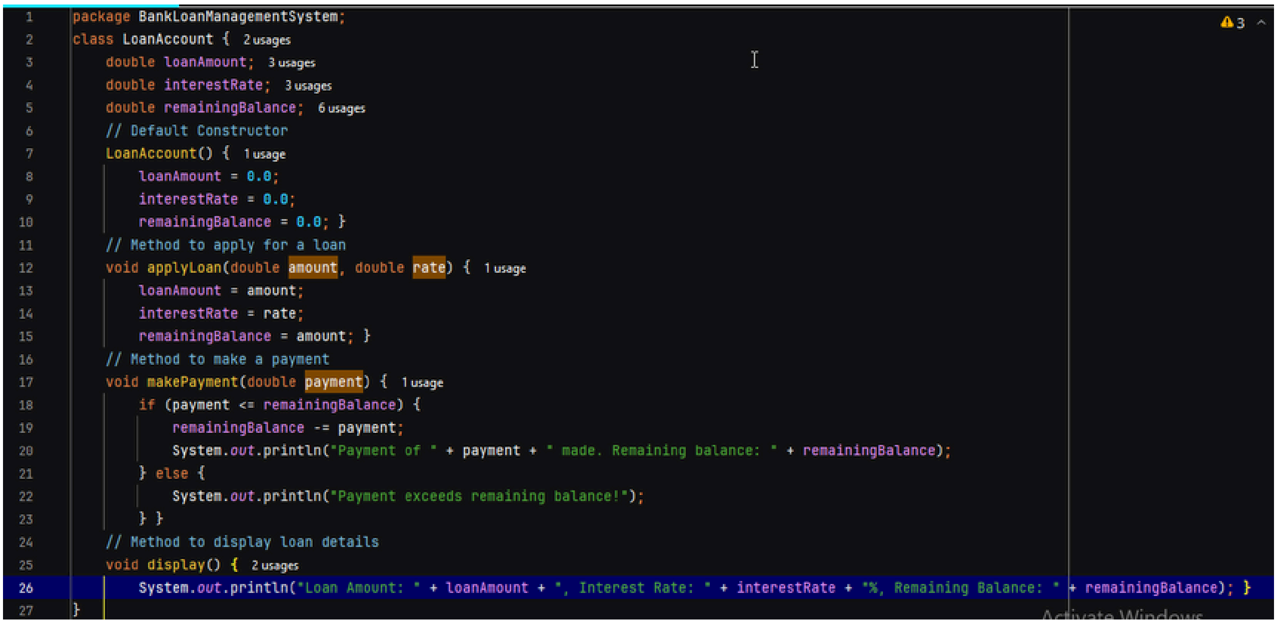
- ***Code Explanation:***
    1. **Class:** Patient has attributes (name, age, disease) and methods (addPatient, updateDisease, display).
    2. **Object:** p1 is an object of the Patient class.
    3. **Methods:**
        - **Add Patient:** Sets patient details.
        - **updateDisease:** Updates the disease.
        - **display:** Shows patient details.

## Q2. BANK LOAN MANAGEMENT SYSTEM

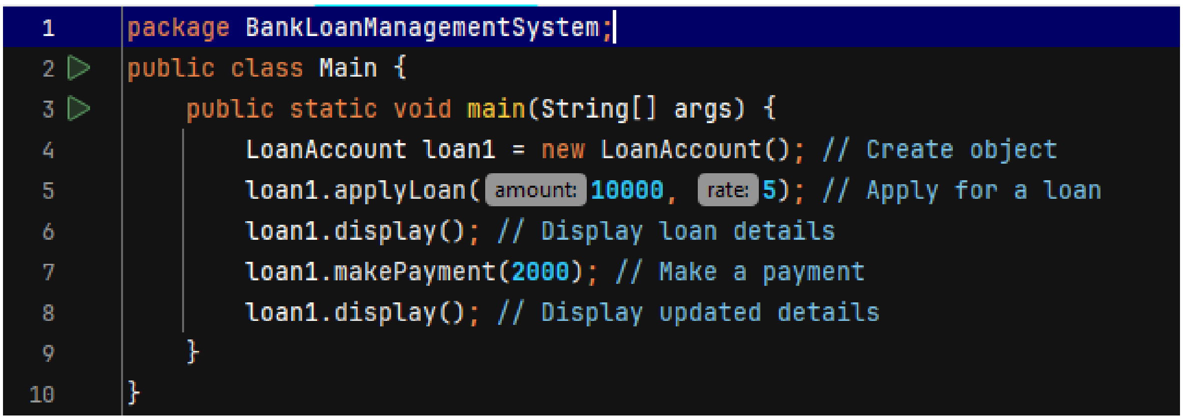
**Problem:** Create a LoanAccount class to manage customer loans. The class should allow:

1. Applying for a loan with details like loan amount and interest rate.
2. Making a payment to reduce the loan balance.
3. Displaying loan details including the remaining balance.

- ***Program:***

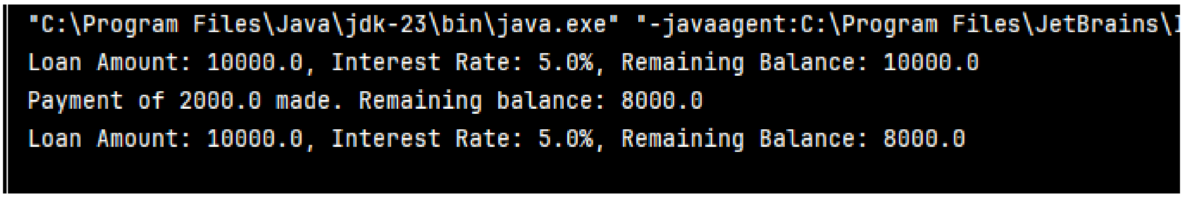


```
package BankLoanManagementSystem;
class LoanAccount {  2 usages
    double loanAmount;  3 usages
    double interestRate;  3 usages
    double remainingBalance;  6 usages
    // Default Constructor
    LoanAccount() {  1 usage
        loanAmount = 0.0;
        interestRate = 0.0;
        remainingBalance = 0.0; }
    // Method to apply for a loan
    void applyLoan(double amount, double rate) {  1 usage
        loanAmount = amount;
        interestRate = rate;
        remainingBalance = amount; }
    // Method to make a payment
    void makePayment(double payment) {  1 usage
        if (payment <= remainingBalance) {
            remainingBalance -= payment;
            System.out.println("Payment of " + payment + " made. Remaining balance: " + remainingBalance);
        } else {
            System.out.println("Payment exceeds remaining balance!");
        } }
    // Method to display loan details
    void display() {  2 usages
        System.out.println("Loan Amount: " + loanAmount + ", Interest Rate: " + interestRate + "%, Remaining Balance: " + remainingBalance); }
}
```



```java
package BankLoanManagementSystem;
public class Main {
    public static void main(String[] args) {
        LoanAccount loan1 = new LoanAccount(); // Create object
        loan1.applyLoan(amount: 10000, rate: 5); // Apply for a loan
        loan1.display(); // Display loan details
        loan1.makePayment(2000); // Make a payment
        loan1.display(); // Display updated details
    }
}
```

- ***Output:***



```
"C:\Program Files\Java\jdk-23\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\I
Loan Amount: 10000.0, Interest Rate: 5.0%, Remaining Balance: 10000.0
Payment of 2000.0 made. Remaining balance: 8000.0
Loan Amount: 10000.0, Interest Rate: 5.0%, Remaining Balance: 8000.0
```

- ***Code Explanation:***

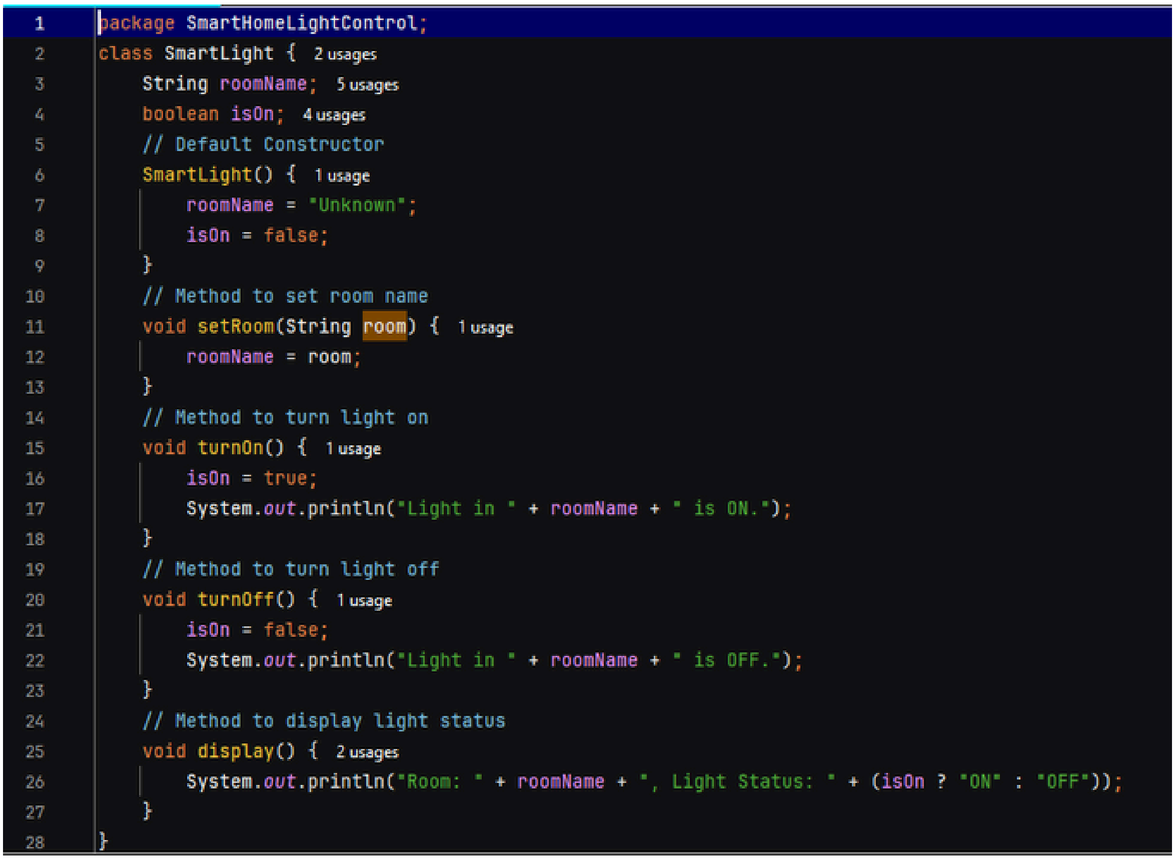
1. **Class**: LoanAccount has attributes (loanAmount, interestRate, remainingBalance) and methods (applyLoan, makePayment, display).
2. **Object**: loan1 is an object of the LoanAccount class.
3. **Methods**:
4. **applyLoan**: Sets loan details.
5. **makePayment**: Reduces the remaining balance.
6. **display**: Shows loan details.

## Q3. SMART HOME LIGHT CONTROL SYSTEM

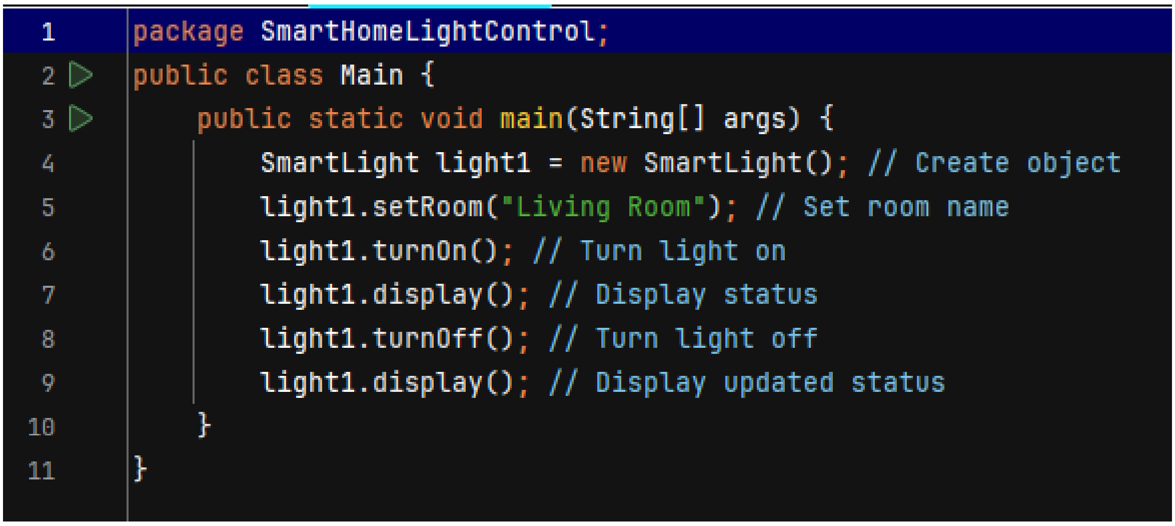
**Problem:** Create a SmartLight class to control lights in different rooms. The class should allow:

1. Turning lights on and off.
2. Checking the current status of the light.
3. Displaying light settings.

- ***Program:***

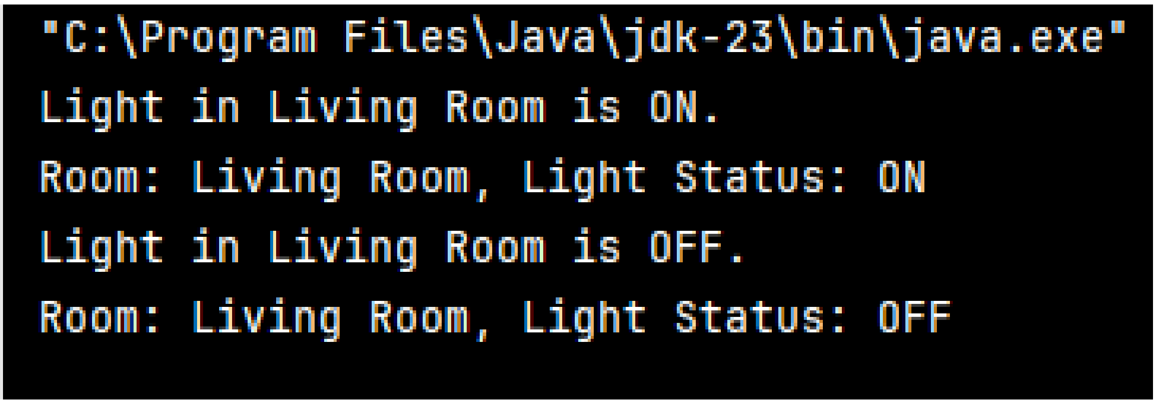


```java
package SmartHomeLightControl;
class SmartLight {
    String roomName;
    boolean isOn;
    // Default Constructor
    SmartLight() {
        roomName = "Unknown";
        isOn = false;
    }
    // Method to set room name
    void setRoom(String room) {
        roomName = room;
    }
    // Method to turn light on
    void turnOn() {
        isOn = true;
        System.out.println("Light in " + roomName + " is ON.");
    }
    // Method to turn light off
    void turnOff() {
        isOn = false;
        System.out.println("Light in " + roomName + " is OFF.");
    }
    // Method to display light status
    void display() {
        System.out.println("Room: " + roomName + ", Light Status: " + (isOn ? "ON" : "OFF"));
    }
}
```



```java
package SmartHomeLightControl;
public class Main {
    public static void main(String[] args) {
        SmartLight light1 = new SmartLight(); // Create object
        light1.setRoom("Living Room"); // Set room name
        light1.turnOn(); // Turn light on
        light1.display(); // Display status
        light1.turnOff(); // Turn light off
        light1.display(); // Display updated status
    }
}
```

- ***Output:***



```
"C:\Program Files\Java\jdk-23\bin\java.exe"
Light in Living Room is ON.
Room: Living Room, Light Status: ON
Light in Living Room is OFF.
Room: Living Room, Light Status: OFF
```

- ***Code Explanation:***

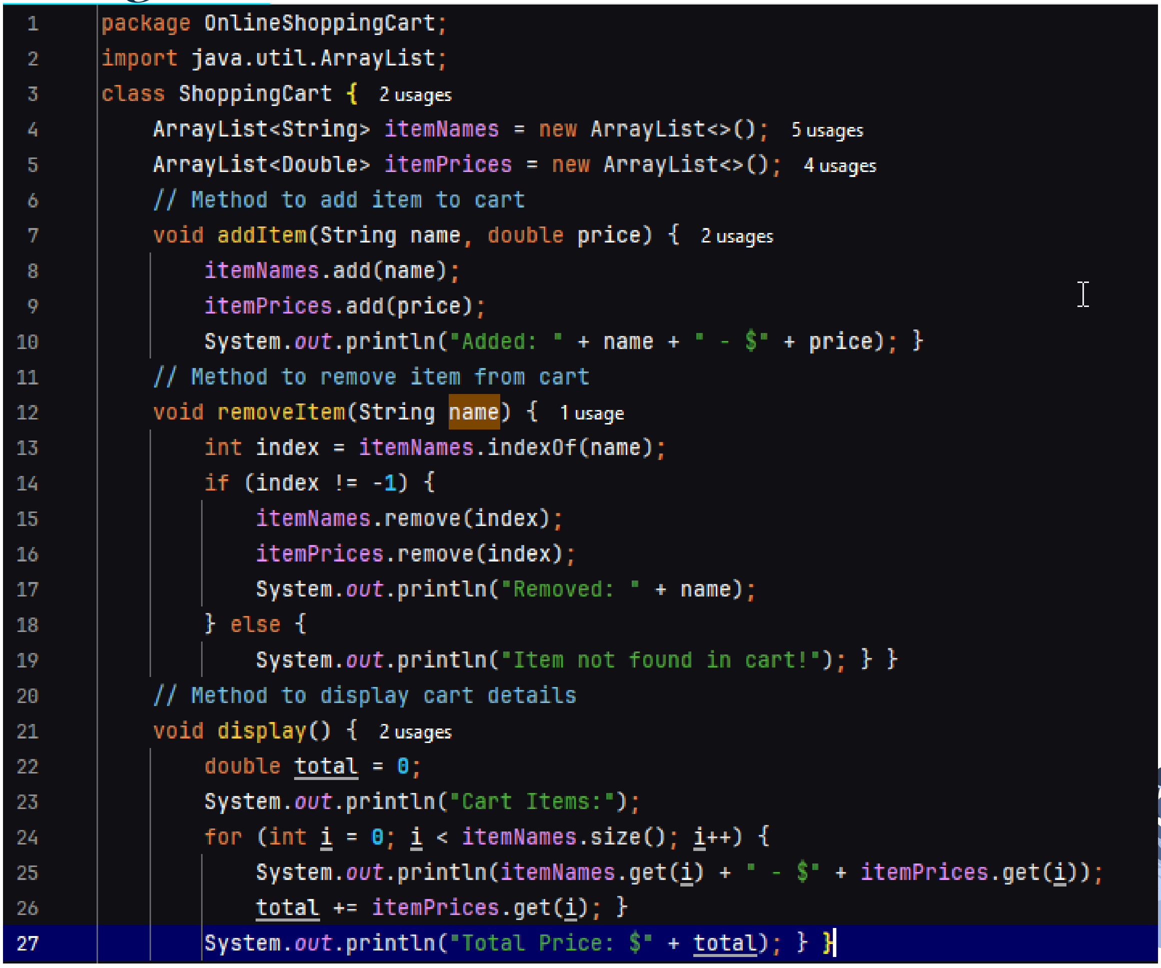
1. **Class:** SmartLight has attributes (roomName, isOn) and methods (setRoom, turnOn, turnOff, display).
2. **Object**: light1 is an object of the SmartLight class.
3. **Methods**:
4. **setRoom**: Sets the room name.
5. **turnOn**: Turns the light on.
6. **turnOff**: Turns the light off.
7. **display**: Shows the light status.

## Q4. ONLINE SHOPPING CART SYSTEM

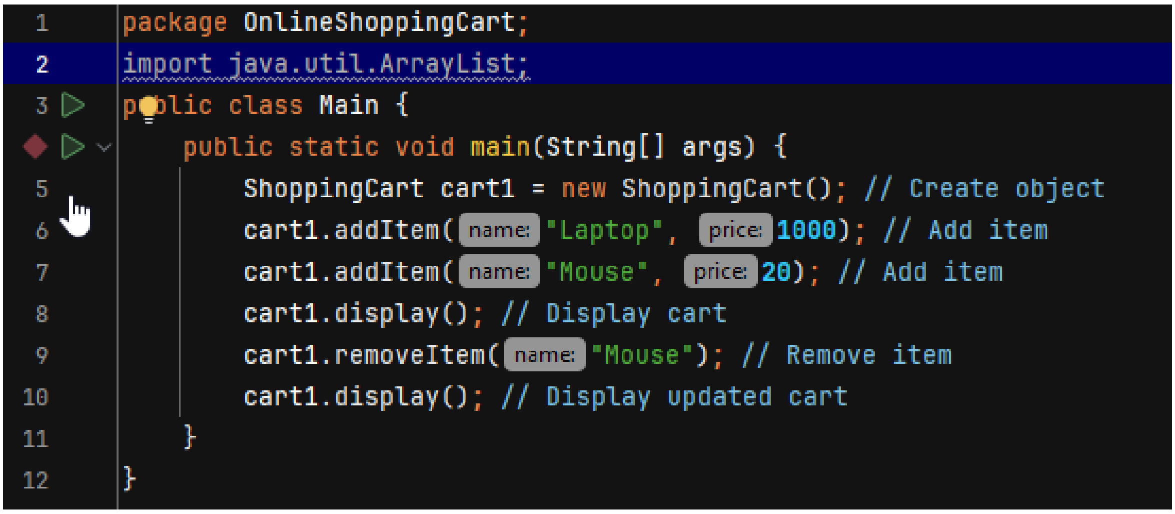
**Problem:** Create a ShoppingCart class to manage shopping carts. The class should allow:

1. Adding items to the cart with item Name and price.
2. Removing items from the cart.
3. Displaying the cart details including the total price.

- ***Program:***

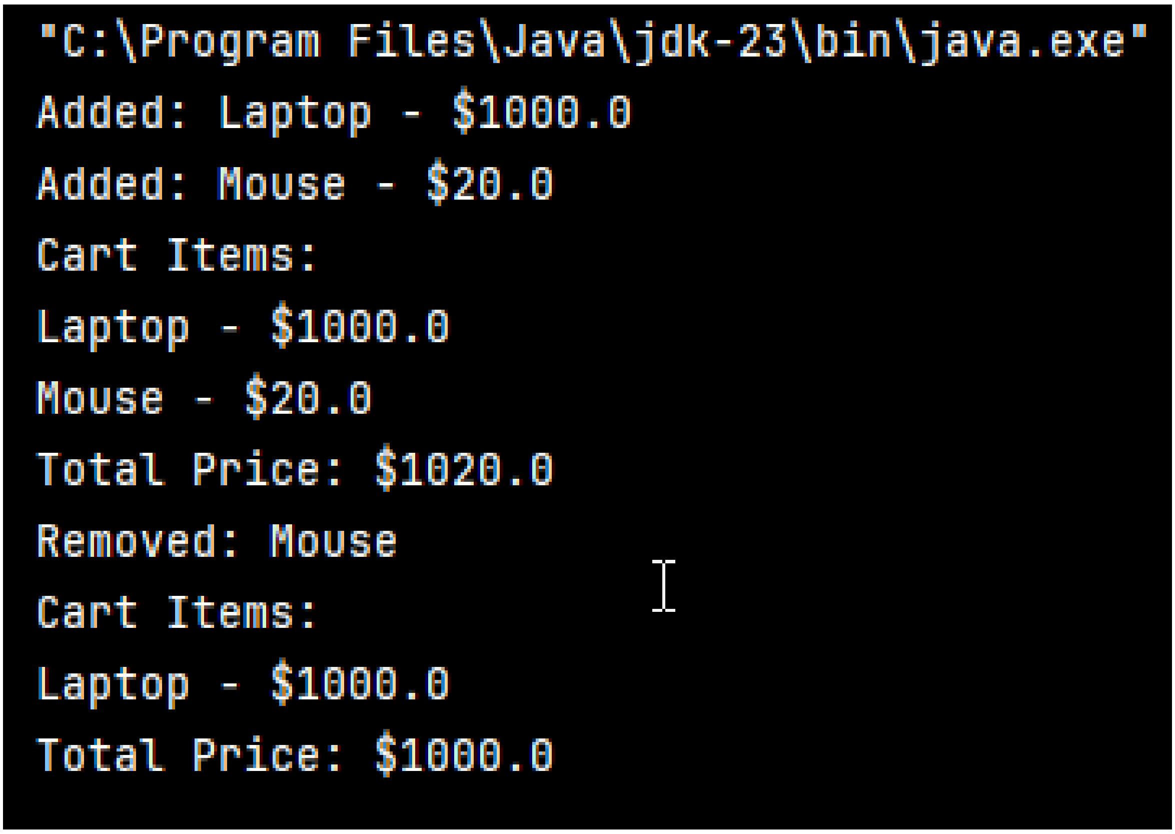


```java
package OnlineShoppingCart;
import java.util.ArrayList;
class ShoppingCart {  2 usages
    ArrayList<String> itemNames = new ArrayList<>();  5 usages
    ArrayList<Double> itemPrices = new ArrayList<>();  4 usages
    // Method to add item to cart
    void addItem(String name, double price) {  2 usages
        itemNames.add(name);
        itemPrices.add(price);
        System.out.println("Added: " + name + " - $" + price); }
    // Method to remove item from cart
    void removeItem(String name) {  1 usage
        int index = itemNames.indexOf(name);
        if (index != -1) {
            itemNames.remove(index);
            itemPrices.remove(index);
            System.out.println("Removed: " + name);
        } else {
            System.out.println("Item not found in cart!"); } }
    // Method to display cart details
    void display() {  2 usages
        double total = 0;
        System.out.println("Cart Items:");
        for (int i = 0; i < itemNames.size(); i++) {
            System.out.println(itemNames.get(i) + " - $" + itemPrices.get(i));
            total += itemPrices.get(i); }
        System.out.println("Total Price: $" + total); } }
```



```java
package OnlineShoppingCart;
import java.util.ArrayList;
public class Main {
    public static void main(String[] args) {
        ShoppingCart cart1 = new ShoppingCart(); // Create object
        cart1.addItem( name: "Laptop",  price: 1000); // Add item
        cart1.addItem( name: "Mouse",  price: 20); // Add item
        cart1.display(); // Display cart
        cart1.removeItem( name: "Mouse"); // Remove item
        cart1.display(); // Display updated cart
    }
}
```

- ***Output:***



```
"C:\Program Files\Java\jdk-23\bin\java.exe"
Added: Laptop - $1000.0
Added: Mouse - $20.0
Cart Items:
Laptop - $1000.0
Mouse - $20.0
Total Price: $1020.0
Removed: Mouse
Cart Items:
Laptop - $1000.0
Total Price: $1000.0
```

- ***Code Explanation:***

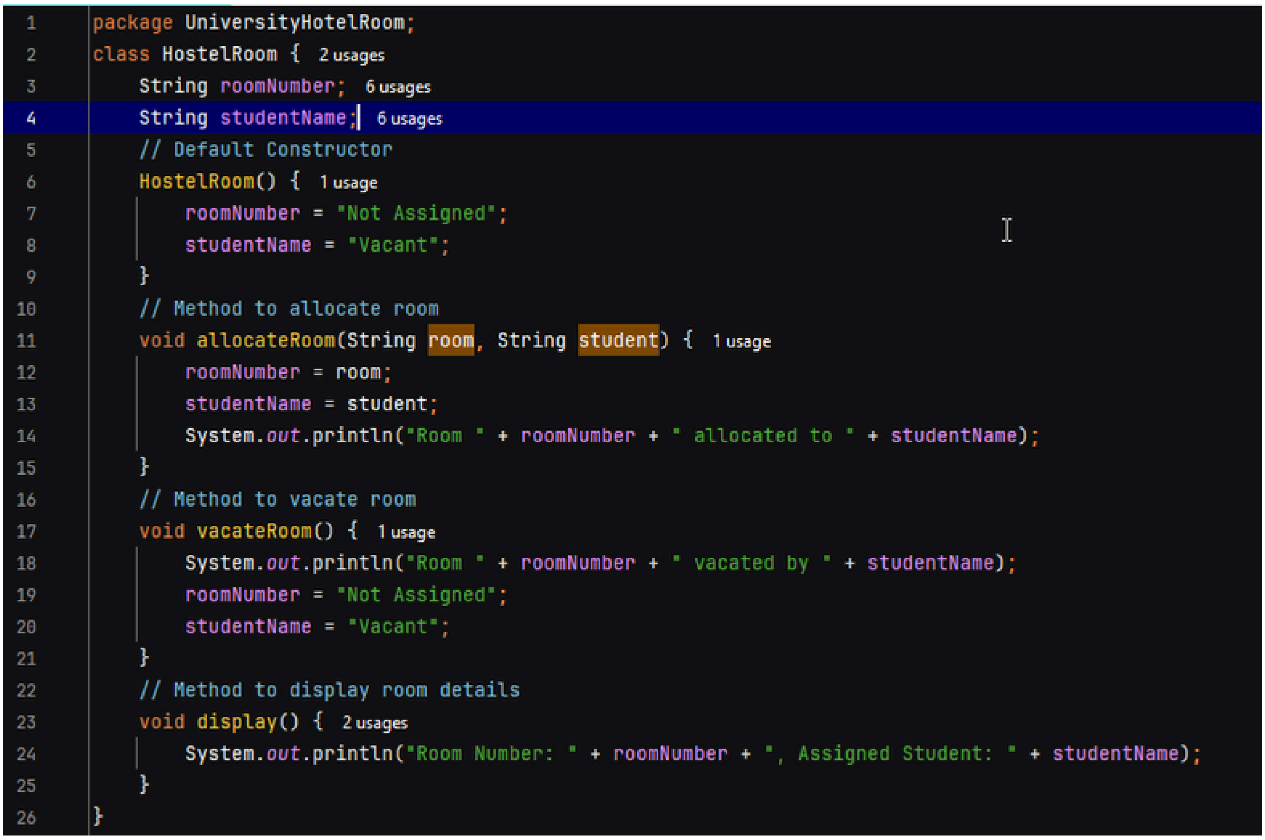
1. **Class:** ShoppingCart uses ArrayList to store itemNames and itemPrices.
2. **Methods**:
3. **addItem**: Adds an item to the cart.
4. **removeItem**: Removes an item from the cart.
5. **display**: Shows all items and the total price.

# Q5. UNIVERSITY HOSTEL ROOM ALLOCATION SYSTEM

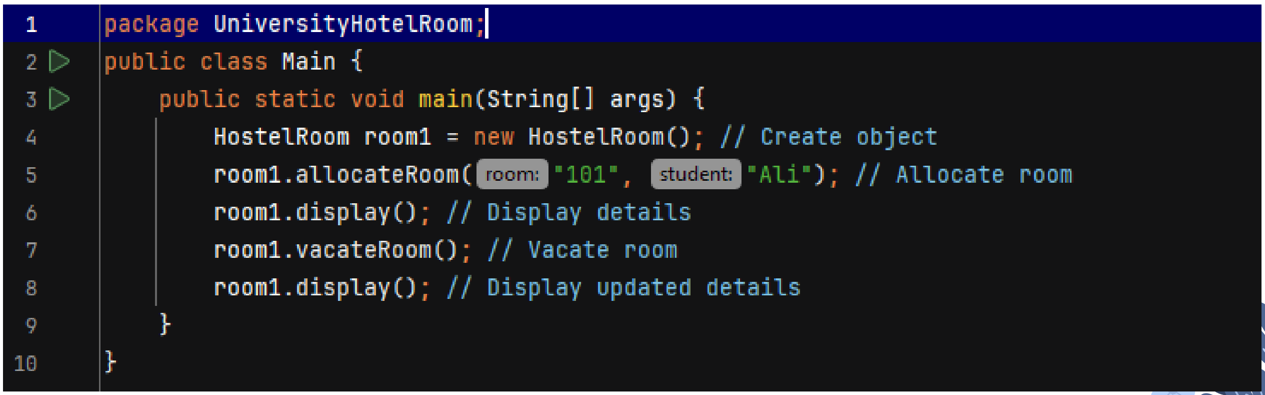
**Problem:** Create a HostelRoom class to manage hostel room allocations. The class should allow:

1. Allocating a room to a student.
2. Vacating a room when a student leaves.
3. Displaying room details including the assigned student's name.

- ***Program:***

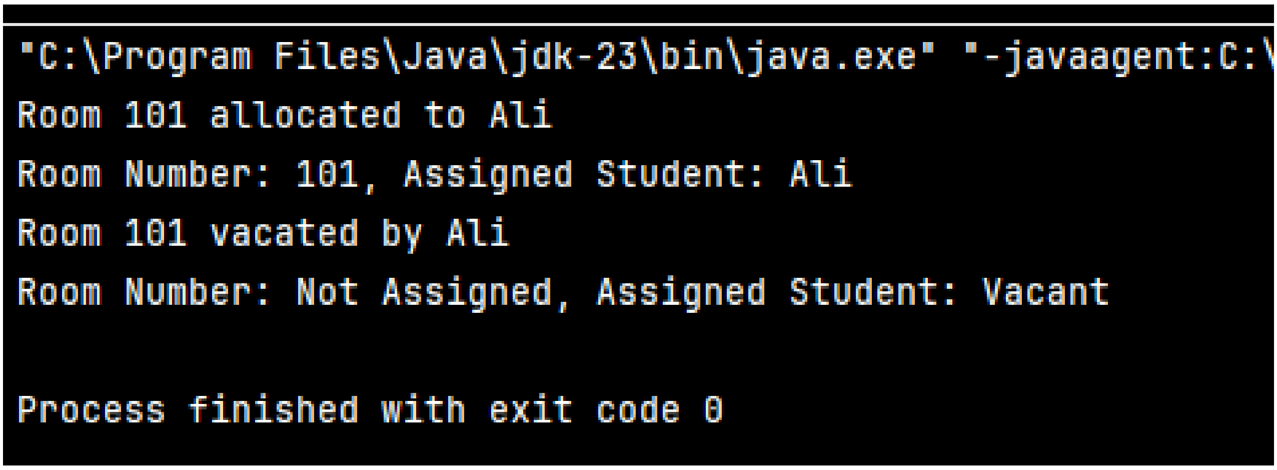


```java
package UniversityHotelRoom;
class HostelRoom {  2 usages
    String roomNumber;  6 usages
    String studentName;  6 usages
    // Default Constructor
    HostelRoom() {  1 usage
        roomNumber = "Not Assigned";
        studentName = "Vacant";
    }
    // Method to allocate room
    void allocateRoom(String room, String student) {  1 usage
        roomNumber = room;
        studentName = student;
        System.out.println("Room " + roomNumber + " allocated to " + studentName);
    }
    // Method to vacate room
    void vacateRoom() {  1 usage
        System.out.println("Room " + roomNumber + " vacated by " + studentName);
        roomNumber = "Not Assigned";
        studentName = "Vacant";
    }
    // Method to display room details
    void display() {  2 usages
        System.out.println("Room Number: " + roomNumber + ", Assigned Student: " + studentName);
    }
}
```



```java
package UniversityHotelRoom;
public class Main {
    public static void main(String[] args) {
        HostelRoom room1 = new HostelRoom(); // Create object
        room1.allocateRoom( room: "101", student: "Ali"); // Allocate room
        room1.display(); // Display details
        room1.vacateRoom(); // Vacate room
        room1.display(); // Display updated details
    }
}
```

- ***Output:***



```
"C:\Program Files\Java\jdk-23\bin\java.exe" "-javaagent:C:\
Room 101 allocated to Ali
Room Number: 101, Assigned Student: Ali
Room 101 vacated by Ali
Room Number: Not Assigned, Assigned Student: Vacant

Process finished with exit code 0
```

- ***Code Explanation:***

1. **Class:** HostelRoom has attributes (roomNumber, studentName) and methods (allocateRoom, vacateRoom, display).
2. **Methods**:
3. **AllocateRoom**: Assigns a room to a student.
4. **Vacate Room:** Marks the room as vacant.
5. **display**: Shows room details.