

```
public class BookEqualityDemo {

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

        public Book(String title, String author) {
            this.title = title;
            this.author = author;
        }

        @Override
        public boolean equals(Object obj) {
            if (this == obj) return true;
            if (obj == null || getClass() != obj.getClass()) return false;
            Book other = (Book) obj;
            return title.equals(other.title) &&
author.equals(other.author);
        }

        @Override
        public String toString() {
            return "\"" + title + "\" by " + author;
        }
    }

    public static void main(String[] args) {
        Book book1 = new Book("Clean Code", "Robert C. Martin");
        Book book2 = new Book("Clean Code", "Robert C. Martin");
        Book book3 = book1;

        System.out.println("book1 == book2: " + (book1 == book2));
// false
        System.out.println("book1.equals(book2): " + book1.equals(book2));
// true

        System.out.println("book1 == book3: " + (book1 == book3));
// true
        System.out.println("book1.equals(book3): " + book1.equals(book3));
// true
    }
}
```

```
}  
}
```

```
PS E:\JAVA PROGRAMS\steparyansingh\year2\oops\week9\lab-work> java BookEqualityDemo  
book1 == book2: false  
book1.equals(book2): true  
book1 == book3: true  
book1.equals(book3): true
```

```
class Address {  
    String city;  
  
    public Address(String city) {  
        this.city = city;  
    }  
  
    // Deep copy constructor  
    public Address(Address other) {  
        this.city = other.city;  
    }  
  
    @Override  
    public String toString() {  
        return "Address [City: " + city + "];"  
    }  
}  
  
class Person implements Cloneable {  
    String name;  
    Address address;
```

```

public Person(String name, Address address) {
    this.name = name;
    this.address = address;
}

// Shallow clone
public Person shallowClone() throws CloneNotSupportedException {
    return (Person) super.clone();
}

// Deep clone
public Person deepClone() throws CloneNotSupportedException {
    Person cloned = (Person) super.clone();
    cloned.address = new Address(this.address); // new Address object
    return cloned;
}

@Override
public String toString() {
    return "Person [Name: " + name + ", " + address + "];"
}
}

public class PersonCloneDemo {
    public static void main(String[] args) throws
CloneNotSupportedException {
        Address addr = new Address("Chennai");
        Person original = new Person("Ravi", addr);

        Person shallow = original.shallowClone();
        Person deep = original.deepClone();

        System.out.println("Before modification:");
        System.out.println("Original: " + original);
        System.out.println("Shallow Clone: " + shallow);
        System.out.println("Deep Clone: " + deep);

        // Modify original's address
        original.address.city = "Bangalore";
    }
}

```

```

        System.out.println("\nAfter modifying original's address:");
        System.out.println("Original: " + original);
        System.out.println("Shallow Clone: " + shallow); // reflects
change
        System.out.println("Deep Clone: " + deep);        // remains
unchanged
    }
}

```

```

● PS E:\JAVA PROGRAMS\steparyansingh\year2\oops\week9\lab-work> java PersonCloneDemo
Before modification:
Original: Person [Name: Ravi, Address [City: Chennai]]
Shallow Clone: Person [Name: Ravi, Address [City: Chennai]]
Deep Clone: Person [Name: Ravi, Address [City: Chennai]]

After modifying original's address:
Original: Person [Name: Ravi, Address [City: Bangalore]]
Shallow Clone: Person [Name: Ravi, Address [City: Bangalore]]
Deep Clone: Person [Name: Ravi, Address [City: Chennai]]

```

```

public class CarInfoDisplay {

    static class Car {
        private String brand;
        private String model;
        private double price;

        public Car(String brand, String model, double price) {
            this.brand = brand;
            this.model = model;
            this.price = price;
        }

        @Override
        public String toString() {

```

```

        return "Car [Brand: " + brand + ", Model: " + model + ",
Price: ₹" + price + "]\n";
    }
}

public static void main(String[] args) {
    Car car = new Car("Tata", "Harrier", 2150000.00);

    // Print object using toString()
    System.out.println("Car Details: " + car);

    // Print class name using getClass().getName()
    System.out.println("Class Name: " + car.getClass().getName());
}
}

```

```

● PS E:\JAVA PROGRAMS\steparyansingh\year2\oops\week9\lab-work> java CarInfoDisplay
Car Details: Car [Brand: Tata, Model: Harrier, Price: ?2150000.0]
Class Name: CarInfoDisplay$Car

```

```

import java.util.HashSet;
import java.util.Objects;

public class StudentHashSetDemo {

    static class Student {
        private int id;
        private String name;

        public Student(int id, String name) {
            this.id = id;
            this.name = name;
        }
    }
}

```

```
@Override
public boolean equals(Object obj) {
    if (this == obj) return true;
    if (obj == null || getClass() != obj.getClass()) return false;
    Student other = (Student) obj;
    return this.id == other.id;
}

@Override
public int hashCode() {
    return Objects.hash(id);
}

@Override
public String toString() {
    return "Student [ID: " + id + ", Name: " + name + "]";
}
}

public static void main(String[] args) {
    HashSet<Student> students = new HashSet<>();

    Student s1 = new Student(101, "Aarav");
    Student s2 = new Student(102, "Bhavna");
    Student s3 = new Student(101, "Aarav"); // Same ID as s1

    students.add(s1);
    students.add(s2);
    students.add(s3); // Should be ignored due to same ID

    System.out.println("Students in HashSet:");
    for (Student s : students) {
        System.out.println(s);
    }
}
}
```

```
PS E:\JAVA PROGRAMS\steparyansingh\year2\oops\week9\lab-work> java StudentHashSetDemo
Students in HashSet:
Student [ID: 101, Name: Aarav]
Student [ID: 102, Name: Bhavna]
PS E:\JAVA PROGRAMS\steparyansingh\year2\oops\week9\lab-work>
```