

```
import java.util.ArrayList;
import java.util.List;

class Book implements Cloneable {
    private String title;

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

    public void setTitle(String title) {
        this.title = title;
    }

    public String getTitle() {
        return title;
    }

    @Override
    protected Book clone() {
        return new Book(this.title);
    }

    @Override
    public String toString() {
        return "Book{title='" + title + "'}";
    }
}

class Library implements Cloneable {
    private List<Book> books;

    public Library(List<Book> books) {
        this.books = books;
    }

    public List<Book> getBooks() {
        return books;
    }
}
```

```

// Shallow clone: copies reference to the same book list
protected Library shallowClone() throws CloneNotSupportedException {
    return (Library) super.clone();
}

// Deep clone: clones each Book object individually
protected Library deepClone() {
    List<Book> clonedBooks = new ArrayList<>();
    for (Book book : books) {
        clonedBooks.add(book.clone());
    }
    return new Library(clonedBooks);
}

@Override
public String toString() {
    return "Library{books=" + books + "}";
}
}

public class LibraryCloneDemo {
    public static void main(String[] args) throws
CloneNotSupportedException {
        List<Book> originalBooks = new ArrayList<>();
        originalBooks.add(new Book("Java Fundamentals"));
        originalBooks.add(new Book("Operating Systems"));

        Library originalLibrary = new Library(originalBooks);

        Library shallowClonedLibrary = originalLibrary.shallowClone();
        Library deepClonedLibrary = originalLibrary.deepClone();

        // Modify a book in shallow clone
        shallowClonedLibrary.getBooks().get(0).setTitle("Modified Java
Book");

        System.out.println("Original Library (after shallow
modification):");
        System.out.println(originalLibrary);
    }
}

```

```

        System.out.println("Shallow Cloned Library:");
        System.out.println(shallowClonedLibrary);

        System.out.println("Deep Cloned Library:");
        System.out.println(deepClonedLibrary);
    }
}

```

```

PS E:\JAVA PROGRAMS\steparyansingh\year2\oops\week9\assignment> java LibraryCloneDemo
Original Library (after shallow modification):
Library{books=[Book{title='Modified Java Book'}, Book{title='Operating Systems'}]}
Shallow Cloned Library:
Library{books=[Book{title='Modified Java Book'}, Book{title='Operating Systems'}]}
Deep Cloned Library:
Library{books=[Book{title='Java Fundamentals'}, Book{title='Operating Systems'}]}

```

```

public class EmployeeInfoPrinter {

    static class Employee {
        private int id;
        private String name;
        private double salary;

        public Employee(int id, String name, double salary) {
            this.id = id;
            this.name = name;
            this.salary = salary;
        }

        @Override
        public String toString() {
            return "Employee{id=" + id + ", name='" + name + "', salary=" + salary + "}";
        }
    }
}

```

```

public static void main(String[] args) {
    Employee emp1 = new Employee(101, "Alice", 55000.0);
    Employee emp2 = new Employee(102, "Bob", 62000.0);
    Employee emp3 = new Employee(103, "Charlie", 48000.0);

    Employee[] employees = {emp1, emp2, emp3};

    for (Employee emp : employees) {
        System.out.println("Object Details: " + emp.toString());
        System.out.println("Class Name: " + emp.getClass().getName());
        System.out.println("-----");
    }
}
}

```

PS E:\JAVA PROGRAMS\steparyansingh\year2\oops\week9\assignment> java EmployeeInfoPrinter

```

-----
Object Details: Employee{id=102, name='Bob', salary=62000.0}
Class Name: EmployeeInfoPrinter$Employee
-----
Object Details: Employee{id=103, name='Charlie', salary=48000.0}
Class Name: EmployeeInfoPrinter$Employee
-----

```

```

public class ProductEqualityDemo {

    static class Product {
        private int productId;
        private String productName;

        public Product(int productId, String productName) {
            this.productId = productId;
            this.productName = productName;
        }

        @Override
    }
}

```

```

    public boolean equals(Object obj) {
        if (this == obj) return true;
        if (obj == null || getClass() != obj.getClass()) return false;
        Product other = (Product) obj;
        return this.productId == other.productId;
    }

    @Override
    public String toString() {
        return "Product{id=" + productId + ", name='" + productName +
        "'}";
    }
}

public static void main(String[] args) {
    Product p1 = new Product(201, "Laptop");
    Product p2 = new Product(201, "Laptop");
    Product p3 = p1;

    System.out.println("p1 == p2: " + (p1 == p2));           // false:
different references
    System.out.println("p1.equals(p2): " + p1.equals(p2));    // true:
same productId
    System.out.println("p1 == p3: " + (p1 == p3));           // true:
same reference
}
}

```

```

PS E:\JAVA PROGRAMS\steparyansingh\year2\oops\week9\assignment> java ProductEqualityDemo
p1 == p2: false
p1.equals(p2): true
p1 == p3: true
PS E:\JAVA PROGRAMS\steparyansingh\year2\oops\week9\assignment> 

```

```
import java.util.HashSet;
```

```
import java.util.Objects;

public class StudentHashSetDemo {

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

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

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

        @Override
        public String toString() {
            return "Student{rollNo=" + rollNo + ", name='" + name + "'}";
        }
    }

    public static void main(String[] args) {
        Student s1 = new Student(301, "Riya");
        Student s2 = new Student(302, "Arjun");
        Student s3 = new Student(301, "Riya"); // same rollNo as s1

        HashSet<Student> studentSet = new HashSet<>();
        studentSet.add(s1);
        studentSet.add(s2);
    }
}
```

```
        studentSet.add(s3); // won't be added due to equals/hashCode match  
        with s1
```

```
        System.out.println("HashSet contents:");  
        for (Student s : studentSet) {  
            System.out.println(s);  
        }  
    }  
}
```

```
PS E:\JAVA PROGRAMS\steparyansingh\year2\oops\week9\assignment> java StudentHashSetDemo  
HashSet contents:  
Student{rollNo=301, name='Riya'}  
Student{rollNo=302, name='Arjun'}  
PS E:\JAVA PROGRAMS\steparyansingh\year2\oops\week9\assignment>
```