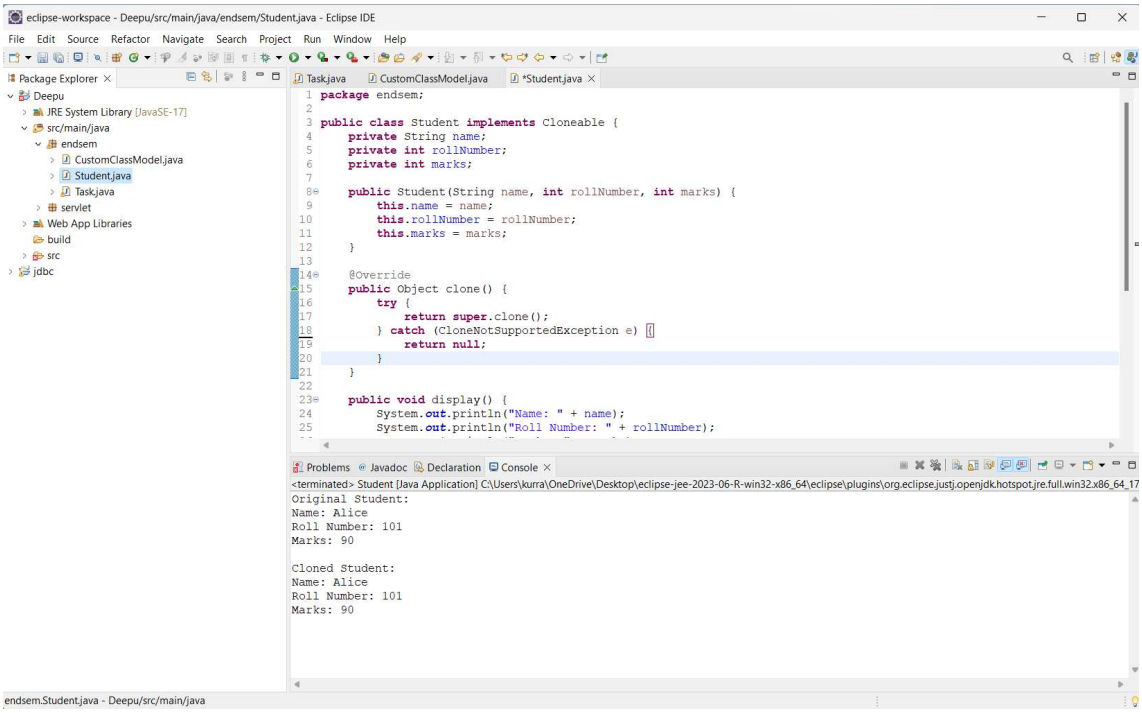
ID:2200031069
Name: K. Deepak

3A)
code:

```
package endsem;

public class Student implements Cloneable {
    private String name;
    private int rollNumber;
    private int marks;

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

    @Override
    public Object clone() {
        try {
            return super.clone();
        } catch (CloneNotSupportedException e) {
            return null;
        }
    }

    public void display() {
        System.out.println("Name: " + name);
        System.out.println("Roll Number: " + rollNumber);
        System.out.println("Marks: " + marks);
    }


    public static void main(String[] args) {

        Student originalStudent = new Student("Alice", 101, 90);


        Student clonedStudent = (Student) originalStudent.clone();

        System.out.println("Original Student:");
        originalStudent.display();

        System.out.println("\nCloned Student:");
        clonedStudent.display();
    }
}
```

Output:



3B)

code:

```java
package endsem;

import java.util.PriorityQueue;

public class Task implements Comparable<Task> {
    private String name;
    private int priority;

    public Task(String name, int priority) {
        this.name = name;
        this.priority = priority;
    }

    public String getName() {
        return name;
    }

    public int getPriority() {
        return priority;
    }

    @Override
    public int compareTo(Task otherTask) {
        // Compare tasks based on their priority
        return Integer.compare(this.priority, otherTask.priority);
    }
}
```

```java
package endsem;

import java.util.PriorityQueue;

public class CustomClassModel {
	public static void main(String[] args) {

		PriorityQueue<Task> taskQueue = new PriorityQueue<>();

		taskQueue.add(new Task("Task 1", 3));
		taskQueue.add(new Task("Task 2", 1));
		taskQueue.add(new Task("Task 3", 2));

		while (!taskQueue.isEmpty()) {
			Task task = taskQueue.poll();
			System.out.println("Processing: " + task.getName() + "
(Priority: " + task.getPriority() + ")");
		}
	}
}
```

Output: