

12 – DAY – TASK (09-08-2024)

1. Implement Abstract class with overloading and overriding.
2. Implement Multiple inheritance with Interface.
3. Show final methods in the class that can't be overridden.

Package1: abstractOverloadingOverriding

File1: Circle.java (Code)

```
package abstractOverloadingOverriding;
```

```
public class Circle extends Shape {
```

```
    @Override
```

```
    public void draw() {
```

```
        // TODO Auto-generated method stub
```

```
        System.out.println("Drawing a circle.");
```

```
    }
```

```
}
```

File2: Rectangle.java (Code)

```
package abstractOverloadingOverriding;
```

```
public class Rectangle extends Shape {
```

```
    @Override
```

```
    public void draw() {
```

```
        // TODO Auto-generated method stub
```

```
        System.out.println("Drawing a rectangle.");
```

```
    }
```

```
}
```

File3: Shape.java (Code)

```
package abstractOverloadingOverriding;
```

```
public abstract class Shape {
```

```
    // Abstract method (overridden by subclasses)
```

```
    public abstract void draw();
```

```
    // Overloaded method
```

```
    public void draw(int size) {
```

```
        System.out.println("Drawing shape of size: " + size);
```

```
    }
```

```
}
```

Package2: multipleInheritance

File1: Colorable.java (Code)

```
package multipleInheritance;
```

```
public interface Colorable {  
    void color();  
}
```

File2: Drawable.java (Code)

```
package multipleInheritance;
```

```
public interface Drawable {  
    void draw();  
}
```

File3: ColorfulCircle.java (Code)

```
package multipleInheritance;
```

```
public class ColorfulCircle implements Drawable, Colorable {
```

```
    @Override
```

```
    public void color() {
```

```
        // TODO Auto-generated method stub
```

```
        System.out.println("Coloring the circle.");
```

```
    }
```

```
    @Override
```

```
    public void draw() {
```

```
        // TODO Auto-generated method stub
```

```
        System.out.println("Drawing the circle.");
```

```
    }
```

```
}
```

Package3: finalMethodClass

File1: Utility.java (Code)

```
package finalMethodClass;
```

```
public class Utility {
```

```
    public final void printMessage() {
```

```
        System.out.println("This is a final method and cannot be overridden.");
```

```
    }
```

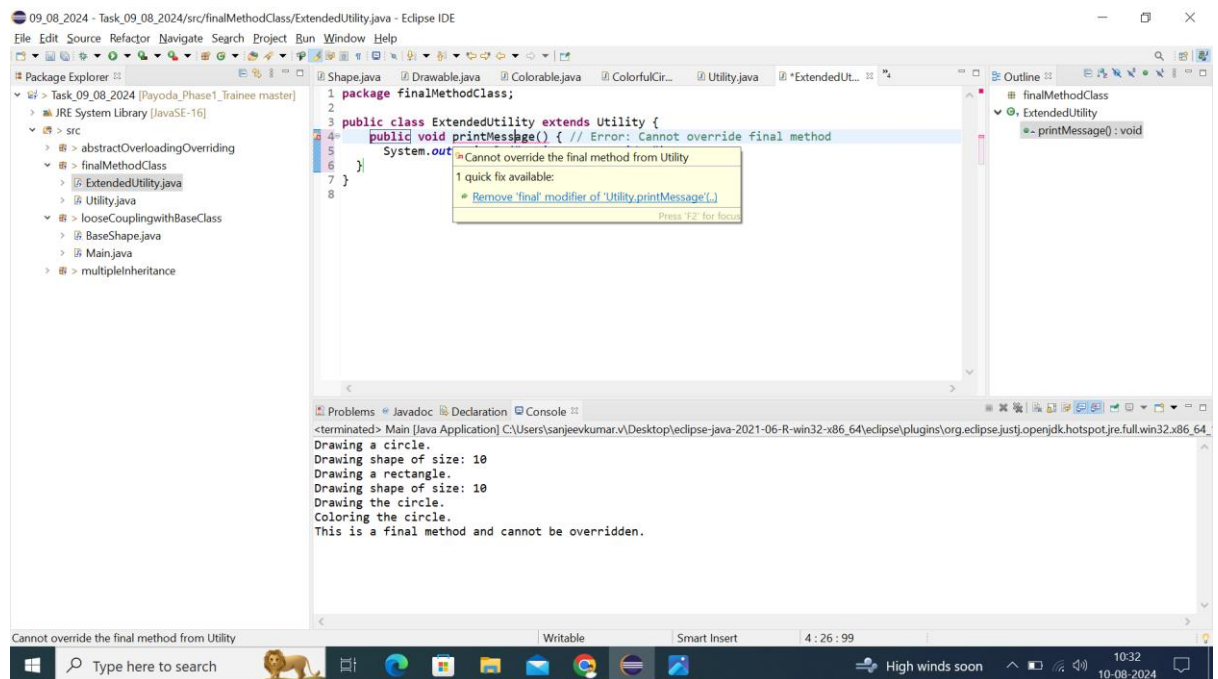
```
}
```

File2: ExtendedUtility.java (Code)

```
package finalMethodClass;
```

```
public class ExtendedUtility extends Utility {  
    void printMessage() { // Error: Cannot override final method  
        System.out.println("Trying to override.");  
    }  
}
```

Error:



Package4: looseCouplingwithBaseClass

File1: BaseShape.java (Code)

```
package looseCouplingwithBaseClass;  
import abstractOverloadingOverriding.Shape;  
//Base class  
class BaseShape {  
    void displayShape(Shape shape) {  
        shape.draw();  
        shape.draw(10); // Overloaded method call  
    }  
}
```

File2: Main.java (Code)

```
package looseCouplingwithBaseClass;
```

```
import abstractOverloadingOverriding.Circle;
import abstractOverloadingOverriding.Rectangle;
import abstractOverloadingOverriding.Shape;
import finalMethodClass.Utility;
import multipleInheritance.ColorfulCircle;

public class Main {
    public static void main(String[] args) {
        BaseShape base = new BaseShape();

        Shape circle = new Circle();
        Shape rectangle = new Rectangle();

        base.displayShape(circle); // Drawing a circle
        base.displayShape(rectangle); // Drawing a rectangle

        ColorfulCircle colorfulCircle = new ColorfulCircle();
        colorfulCircle.draw(); // Drawing a circle
        colorfulCircle.color(); // Coloring the circle

        Utility utility = new Utility();
        utility.printMessage(); // This is a final method and cannot be overridden.
    }
}
```

Output:

