Q1: Answer
⇒

```
package feb1;

class Shape {
    private String color;

    public Shape(String color) {
        this.color = color;
    }

    @Override
    public String toString() {
        return "Shape[color=" + color + "]";
    }

    public double getArea() {
        System.out.println("Invalid");
        return 0;
    }
}

class Rectangle extends Shape {
    private int length, width;

    public Rectangle(String color, int length, int width) {
        super(color);
        this.length = length;
        this.width = width;
    }

    @Override
    public String toString() {
        return "Rectangle[length=" + length + ",width=" + width + "," + super.toString() + "]";
    }

    @Override
    public double getArea() {
        return length * width;
    }
}

class Triangle extends Shape {
    private int base, height;
```

```
    public Triangle(String color, int base, int height) {
        super(color);
        this.base = base;
        this.height = height;
    }

    @Override
    public String toString() {
        return "Triangle[base=" + base + ",height=" + height + "," + super.toString() + "]";
    }

    @Override
    public double getArea() {
        return 0.5 * base * height;
    }
}

public class TestShape {
    public static void main(String[] args) {
        Triangle t1 = new Triangle("Green", 10, 5);
        System.out.println(t1);
        System.out.println("Area of triangle is:: " + t1.getArea());
        Rectangle r1 = new Rectangle("Red", 5, 6);
        System.out.println(r1);
        System.out.println("Area of Rectangle is:: " + r1.getArea());

    }
}
```

Output:

```
Triangle[base=10,height=5,Shape[color=Green]]
Area of triangle is:: 25.0
Rectangle[length=5,width=6,Shape[color=Red]]
Area of Rectangle is:: 30.0
```

Q2: Answer

⇒

```
package Feb1Assingment;

class Shape {
    void calculateArea(int length, int breath){
        System.out.println("The area of Rectangle is "+(length*breath));
    }
```

```
    void calculateArea(double base , double height){
        System.out.println("The area of Triangle is "+(0.5*base*height));
    }
    void calculateArea(int radius){
        System.out.println("The area of Circle is "+(Math.PI*radius*radius));
    }
}

public class Q2 {
    public static void main(String[] args) {
    Shape s1= new Shape();
    s1.calculateArea(5);
    s1.calculateArea(5,10);
    s1.calculateArea(10d,5d);
    }

}
```

Output:

```
The area of Circle is 78.53981633974483
The area of Rectangle is 50
The area of Triangle is 25.0
```

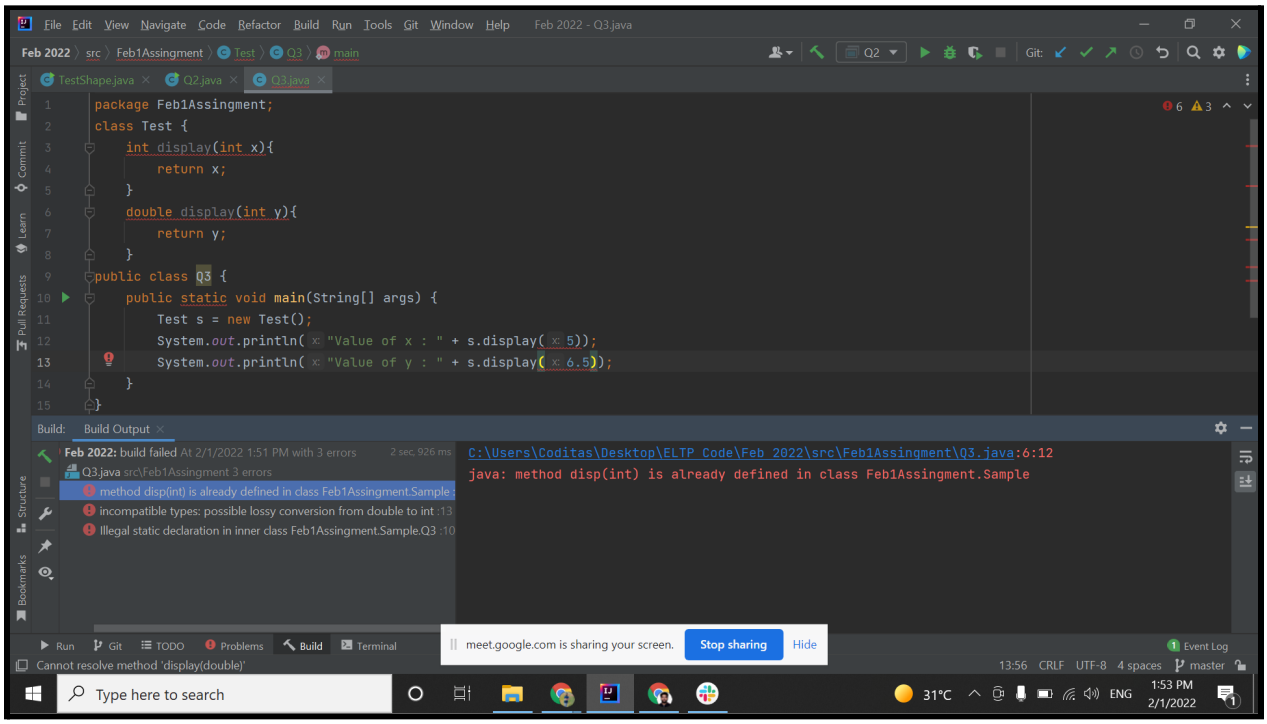
Q3: Answer

- Method overloading cannot be done by changing the return type of methods.
- The most important rule of method overloading is that two overloaded methods must have different parameters.

```
package Feb1Assingment;
class Test {
    int display(int x){
        return x;
    }
    double display(int y){
        return y;
    }
public class Q3 {
    public static void main(String[] args) {
        Test s = new Test();
        System.out.println(Value of x   + s.display(5));
        System.out.println(Value of y   + s.display(6.5));
    }
```

```
}
}
```

Output:



Q4:Answer

⇒

Before Java5, it was not possible to override any method by changing the return type. But now, since Java5, it is possible to override method by changing the return type if subclass overrides any method whose return type is Non-Primitive.

```
package Feb1Assingment;

class base {
    void m1() {
        System.out.println("Void Return Type");
    }

    int m1(int n) {
        System.out.println("Integer Return Type");
        return 0;
    }
}

public class Q4 {
    public static void main(String[] args) {
```

```
        base obj = new base();
        obj.m1();
        obj.m1(7);
    }
}
```

Output:

```
Void Return Type
Integer Return Type
```