

King Saud University
College of Computer and Information Sciences
Department of Computer Science
CSC113 – Computer Programming II – Exception Handling Tutorial Solution– Spring 2018

```
import java.util.Scanner;

public class Test {
    public static int division(int x, int y) throws ArithmeticException{
        return x/y;
    }

    public static void interval(int x, int y) {
        if(x > y)
            throw new IllegalArgumentException();
        for(int i = x; i <= y; i++) {
            System.out.println(i);
        }
    }

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        try {
            int d = division(input.nextInt(), input.nextInt());
            System.out.println(d);
        }
        catch(ArithmeticException e) {
            System.out.println("Exception caught: Division by zero.");
        }
        try {
            interval(input.nextInt(), input.nextInt());
        }
        catch(IllegalArgumentException e) {
            System.out.println("The first number should be less than the
second number.");
        }
        input.close();
    }
}

public class Triangle {
    private int side1;
    private int side2;
    private int side3;

    public Triangle(int side1, int side2, int side3) {
        if (side1 <= 0 || side2 <= 0 || side3 <= 0)
            throw new IllegalArgumentException("Sides can only be positive
numbers");
        this.side1 = side1;
        this.side2 = side2;
        this.side3 = side3;
    }
}
```

King Saud University
College of Computer and Information Sciences
Department of Computer Science
CSC113 – Computer Programming II – Exception Handling Tutorial Solution– Spring 2018

```
public int getSide1() {
    return side1;
}

public void setSide1(int side1) {
    this.side1 = side1;
}

public int getSide2() {
    return side2;
}

public void setSide2(int side2) {
    this.side2 = side2;
}

public int getSide3() {
    return side3;
}

public void setSide3(int side3) {
    this.side3 = side3;
}

public boolean isValidTriangle(){
    if ((side1 + side2 > side3) && (side1 + side3 > side2) && (side3 + side2
> side1))
        return true;
    else
        return false;
}

public String getTriType() throws Exception {
    if (isValidTriangle()) {
        if ((side1 == side2) && (side2 == side3))
            return "equilateral";
        else if ((side1 == side2) || (side1 == side3) || (side2 == side3))
            return "isosceles";
        else
            return "scalene";
    }
    else
        throw new Exception("The sides don't form a valid triangle.");
}
}
```