Assignment: Week 2 Exception Handling

Program 1: Write a small piece of code which shows simple usage of try-catch block with throw and throws keyword.

```
public class _01_throw_and_throws {
    public static void main(String[] args) {
        try{
            printNum(10);
            printNum(200);
        }
        catch(Exception e){
            e.printStackTrace();
        }
    }

    public static void printNum(int n) throws Exception {
       if(n>100){
        throw new Exception("Exception thrown");
       }
       else{
        System.out.println("Number is: "+n);
       }
    }
}
```

Output:

```
Number is: 10
java.lang.Exception: Exception thrown
        at _01_throw_and_throws.printNum(_01_throw_and_throws.java:16)
        at _01_throw_and_throws.main(_01_throw_and_throws.java:7)
```

Program 2: Write code to throw a custom exception when entered number is greater than 100 or less than 0.

```
import java.util.Scanner;
public class _02_CustomException {

    private static class NumberOutOfoundException extends RuntimeException{
        NumberOutOfoundException(String s){
            super(s);
        }
    }
    public static void main(String[] args) {
        System.out.println("Enter a number: ");
        Scanner sc=new Scanner(System.in);
      int num=sc.nextInt();
     try{
        if(isNumOutOfBounds(num)){
            throw new NumberOutOfoundException("Number out of bounds");
        }
        System.out.println("Number is: "+num);
```

```
     }
     catch(NumberOutOfoundException e){
        e.printStackTrace();
     }
      finally{
          sc.close();

      }

    }
    private static boolean isNumOutOfBounds(int num) {
        return num>100 || num<0;
    }
}
```

Output:

```
Enter a number:
105
_02_CustomException$NumberOutOfoundException: Number out of bounds
        at _02_CustomException.main(_02_CustomException.java:16)
```

Program 3: Program to demonstrate chained exceptions

```
import java.util.Scanner;

public class _03_ChainedExceptions {

    private static class ApiException extends RuntimeException {
        ApiException(String m) {
            super(m);
        }
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter two numbers");
        int n = sc.nextInt();
        int m = sc.nextInt();

        try {
            int sum = calcSum(n, m);
            System.out.println("Quotient: " + sum);
        } catch (ArithmeticException e) {
            ApiException apiException = new ApiException("Error occurred while
performing division operation");
            apiException.initCause(e);
            throw apiException;
        }
        finally{
            sc.close();
        }
    }

    private static int calcSum(int a, int b) {
      return a/b;
```

```
    }
}
```

```
Enter two numbers
25
0
Exception in thread "main" _03_ChainedExceptions$ApiException: Error occurred while performing division ope
ration
        at _03_ChainedExceptions.main(_03_ChainedExceptions.java:23)
Caused by: java.lang.ArithmeticException: / by zero
        at _03_ChainedExceptions.calcSum(_03_ChainedExceptions.java:33)
        at _03_ChainedExceptions.main(_03_ChainedExceptions.java:20)
```