Exercises on Exception (SOLUTIONS)

1. The following method will create an empty file. You can supply the file name, including the complete path, as the argument, s.

```
  public static void createFile(String s) throws IOException{
     new File(s).createNewFile();
  }
```

If you do not have permission to create the file, an exception will be thrown. For example, if you try to create a file by calling *createFile("/root/abc.txt")*, an exception will most likely be thrown since most people do not have write permission in /root directory (unless you are logged in as root on a Unix system).

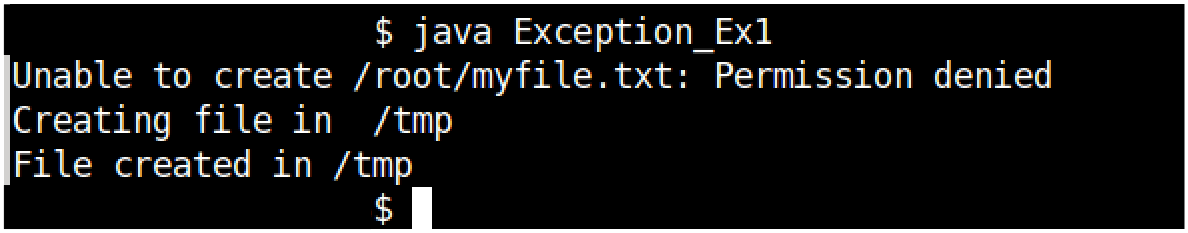
In this exercise, you have to write a program that handles this exception, if it is thrown. One way to handle the exception is to create the file in a location where most people *do* have permission to create a file, such as /tmp. So if the user asks to create a file /root/myfile.txt, an exception should be thrown and a file must be created in /tmp, named myfile.txt.

```
import java.io.*;
class Exception_Ex1 {
   public static void createFile(String s) throws IOException{
      new File(s).createNewFile();
   }
   public static void main(String[] args) {
      String filename = "/root/myfile.txt";

      //try to create the file, catch exception, print appropriate messages
      //YOUR CODE  goes here

      // Execution continues here after the Exception handler is done
      System.out.println("File created in /tmp");
   }
}
```

unless you are logged in as root, you will not have write permission to create /root/myfile.txt



```
$ java Exception_Ex1
Unable to create /root/myfile.txt: Permission denied
Creating file in  /tmp
File created in /tmp
$
```

A SOLUTION

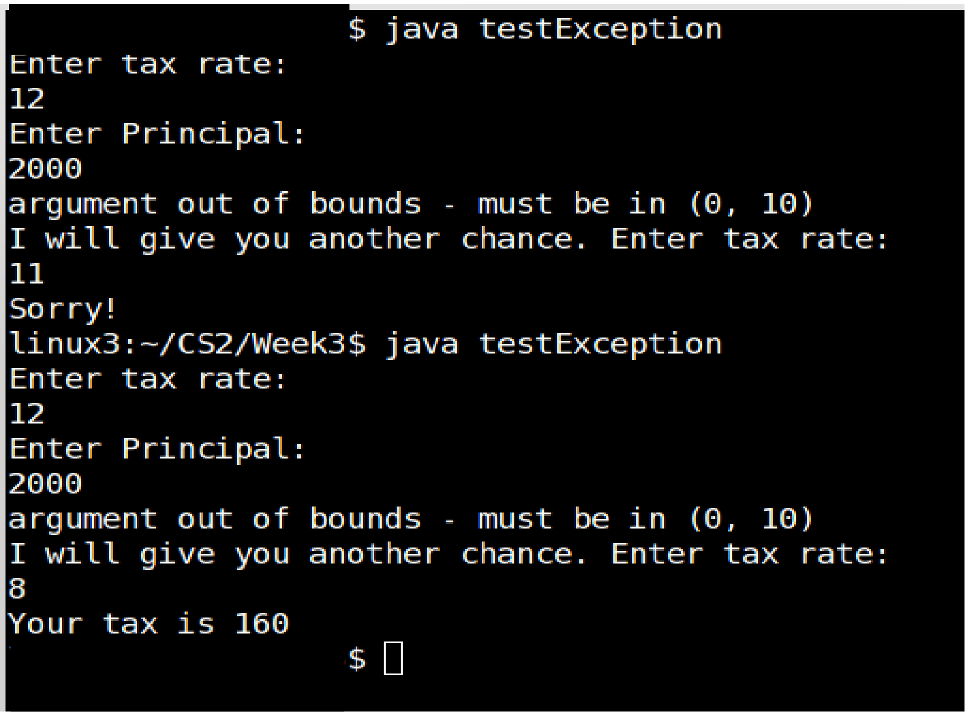
```
import java.io.*;
class Exception_Ex1 {
   public static void createFile(String s) throws IOException{
        new File(s).createNewFile();
    }
    public static void main(String[] args) {
        String filename = "/root/myfile.txt";
        try {
            // Create the file
                createFile(filename);
        } catch (IOException e) {
            // Print out the exception that occurred
            System.out.println("Unable to create "+filename+": "+e.getMessage());
            System.out.println("Creating file in  /tmp");
            try{createFile("/tmp/myfile.txt");
                }
                catch (IOException e2){e2.getMessage();
                }
        }
        // Execution continues here after the Exception handler is done
        System.out.println("File created in /tmp");
    }
}
```

2. In the following exercise you are using your your own exception class. The user inputs an interest rate and a principal amount and a method calculates and displays the interest. If the input rate is beyond a given range, (0,10), the exception is thrown. Your task is to complete the program using the comments and the screenshot of the program being run.

```
import java.util.Scanner;
class myException extends Exception {
   public myException(String s){
            super(s);
   }
}
class testException {
   public static void f(int i, int P) throws myException{
            if (i > 10 || i < 0){
               throw new myException("argument out of bounds - must be in (0, 10)");
            }
            else
               System.out.println("Your tax is " + i*P/100);
   }
   public static void main(String[] args) {
            Scanner sc=new Scanner(System.in);
            System.out.println("Enter tax rate: ");
            int rate=sc.nextInt();
            System.out.println("Enter Principal: ");
            int P=sc.nextInt();
         /** YOUR CODE to call the method f(rate, P), catch any thrown exception. If an exception is thrown, the
user must be given ONE chance to re-input the rate. If the input is no good the second time, the program should
exit with a message. If the input rate is OK, the result of the method call is displayed. The screenshot of the
program run with different input values is shown below
```

Exception class
This is a checked exception

```
  **/
  }
}
```



```
[redacted] $ java testException
Enter tax rate:
12
Enter Principal:
2000
argument out of bounds - must be in (0, 10)
I will give you another chance. Enter tax rate:
11
Sorry!
linux3:~/CS2/Week3$ java testException
Enter tax rate:
12
Enter Principal:
2000
argument out of bounds - must be in (0, 10)
I will give you another chance. Enter tax rate:
8
Your tax is 160
[redacted] $ ▯
```

A SOLUTION

```
import java.util.Scanner;
class myException extends Exception {
   public myException(String s){
         super(s);
   }
}


class testException {

   public static void f(int i, int P) throws myException{
         if (i > 10 || i < 0){
            throw new myException("argument out of bounds - must be in (0, 10)");
         }
         else
            System.out.println("Your tax is " + i*P/100);
   }

   public static void main(String[] args) {
         Scanner sc=new Scanner(System.in);
         System.out.println("Enter tax rate: ");
         int rate=sc.nextInt();
```

```
      System.out.println("Enter Principal: ");
      int P=sc.nextInt();
      try{
        f(rate,P);
      }catch(myException e){
        System.out.println(e.getMessage());
        System.out.println("I will give you another chance. Enter tax rate: ");
        rate=sc.nextInt();
        try{
            f(rate,P);
        }catch(myException e2){
            e2.getMessage();
            System.out.println("Sorry!");
        }
      }
  }
}
```