Name _____ Period_____

1. Refer to the code below,

```
//Assume nextLine() and nextInt() are static methods in
//a class named Scanner that reads a String and an integer
//from the keyboard.

Scanner rdr = new Scanner(System.in);
String str = rdr.nextLine();
int j = rdr.nextInt();

try  {
      System.out.print( str.charAt(j) );

} catch(StringIndexOutOfBoundsException e) {

      System.out.print("Error: " + j);
}
```

(a) What is the output of the code above, given the input below?

```
big mama
2
```

/1

(b) What is the output of the code above, given the input below?

```
big mama
22
```

/1

2. Refer to the code below,

```
//Returns the product of two integers represented as
//strings. If either string cannot be parsed,
//a message indicating why is printed
//If both are not numbers, returns 1.

public static int product(String str1, String str2)  {

int prd = 1;
try  {
      prd*=Integer.parseInt(str1);

     } catch(NumberFormatException e)  {
      <*1>

     }  try  {
      prd*=Integer.parseInt(str2);

     }  catch(NumberFormatException e)  {
      <*2>
     }
      return prd;
}
```

(a) What should replace <*1> in the code above to make it do what the remarks suggest?

/1

(b) What is returned by product ("two", "5")?

/1

3. What is output by the code to the right if the static method called `test()` encounters the following line of code? Assume the test signature includes throws NumberFormatException.

```
int j =
Integer.parseInt("Two
Thousand");
```

```
try{
      test();

}
catch(NumberFormatException e) {
     System.out.println("Error with number
format");

}  catch(RuntimeException e) {
     System.out.println("Error");

}
```

/1

4. If the code designated by <*1> to the right does not throw any exceptions, which of the remaining code sections will execute?

```
try  {
      <*1>

}  catch(RunTimeException e)   {

      <*2>

}  finally  {

      <*3>
}
```

/1

5. The following code blocks could cause errors. Fix the code to prevent an error from occurring

```
class InfiniteLoop {

     public static void test(int i)  {

          for(int i = 1; i > 0; i++){
               System.out.println(i);
          }

     }
}
```

```
class StackOverflow {

     public static void test(int i)
     {
     //Not correct as base condition
     //leads to non-stop recursion if
     //if i is positive

          if (i == 0)
              return;
          else
              test(i++);

      }
}
```

Score _____/23

		/2

6. For each of the following code segments identify the unchecked error that would occur

(a)
```
int[] myArray = {1, 2, 3}
System.out.println(myArray[3]);
```

(b)
```
System.out.println(10/0);
```

(c)
```
String pointer = null;
    if(pointer.equals("this")
        //do something
```

(d)
```
Object x = new Integer(0);
System.out.println((String)x);
```

(e)
```
String s = "Hello";
System.out.println(s.charAt(5));
```

/5

7. For each of the following unchecked errors, write a try-catch block to catch the error.

```
int[] myArray = {1, 2, 3}
System.out.println(myArray[3]);
```

Score _____/23

```
System.out.println(10/0);
```

```
String pointer = null;
if(pointer.equals(“this”)
            //do something
```

```
Object x = new Integer(0);
System.out.println((String)x);
```

```
String s = “Hello”;
System.out.println(s.charAt(5));
```

/10