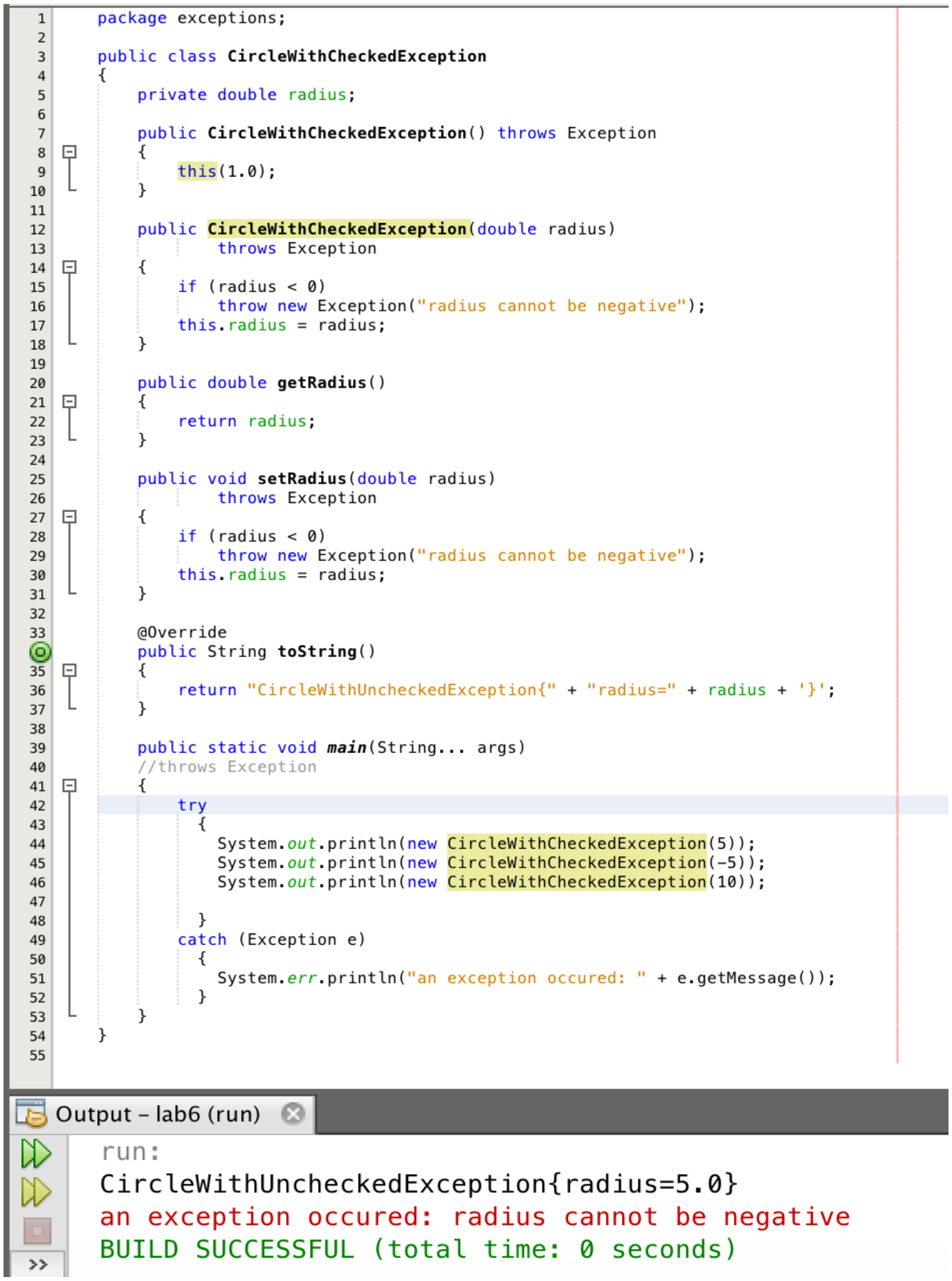
ADSV 2420, Advanced Programming I
Lab – Exceptions and Text Files

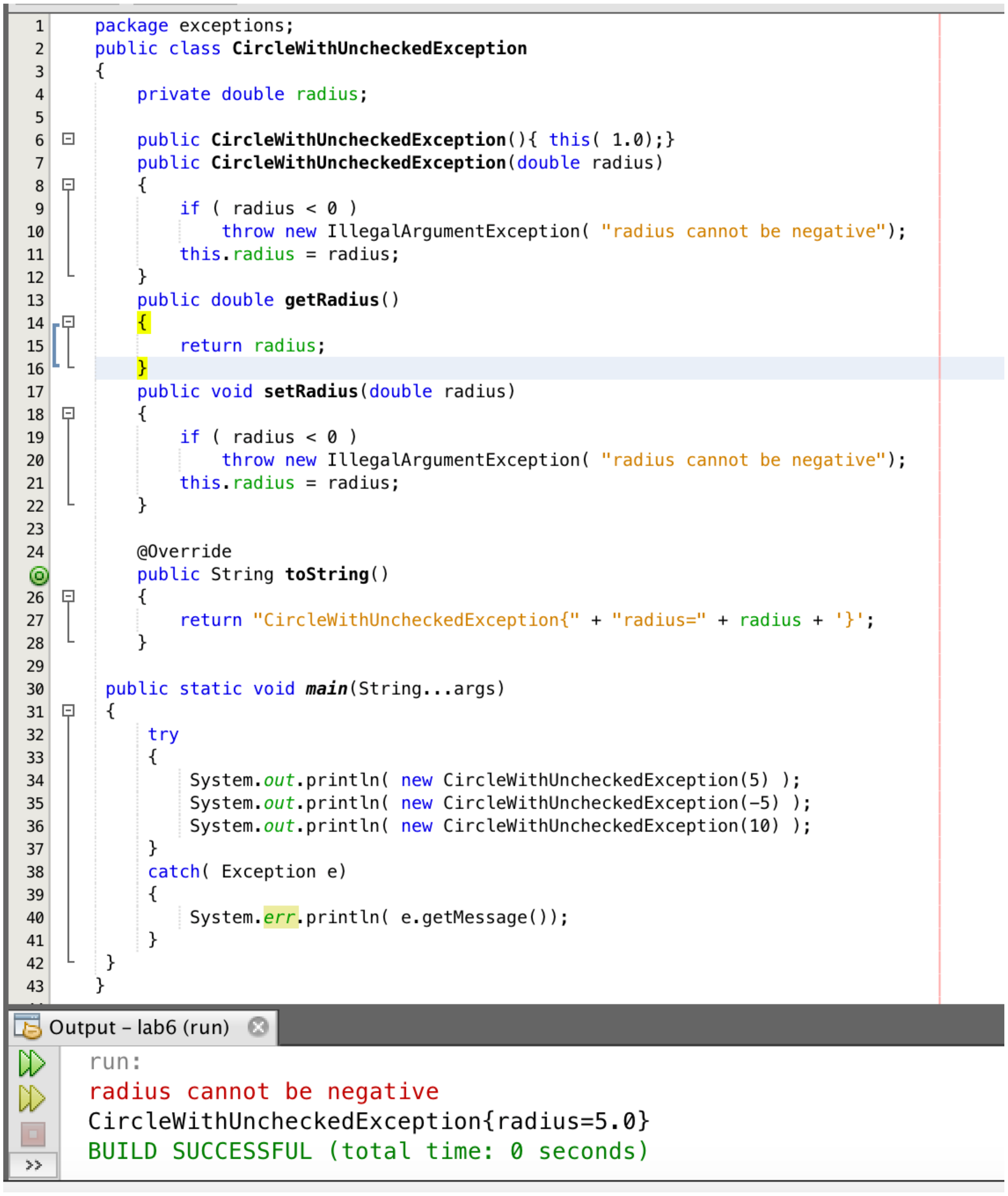
1. Create a new project lab12, create package exceptions and implement the class <u>CircleWithCheckedException</u> .



```java
package exceptions;

public class CircleWithCheckedException
{
    private double radius;

    public CircleWithCheckedException() throws Exception
    {
        this(1.0);
    }

    public CircleWithCheckedException(double radius)
            throws Exception
    {
        if (radius < 0)
            throw new Exception("radius cannot be negative");
        this.radius = radius;
    }

    public double getRadius()
    {
        return radius;
    }

    public void setRadius(double radius)
            throws Exception
    {
        if (radius < 0)
            throw new Exception("radius cannot be negative");
        this.radius = radius;
    }

    @Override
    public String toString()
    {
        return "CircleWithUncheckedException{" + "radius=" + radius + '}';
    }

    public static void main(String... args)
    //throws Exception
    {
        try
        {
            System.out.println(new CircleWithCheckedException(5));
            System.out.println(new CircleWithCheckedException(-5));
            System.out.println(new CircleWithCheckedException(10));

        }
        catch (Exception e)
        {
            System.err.println("an exception occured: " + e.getMessage());
        }
    }
}

```

```
run:
CircleWithUncheckedException{radius=5.0}
an exception occured: radius cannot be negative
BUILD SUCCESSFUL (total time: 0 seconds)
```

2. Copy and paste (refactor) the previous class CircleWithCheckedException into class CircleWithUncheckedException. Observe the the exception IllegalArgumentException is an unchecked exception and we don't have to declare it at headers of methods or constructors.

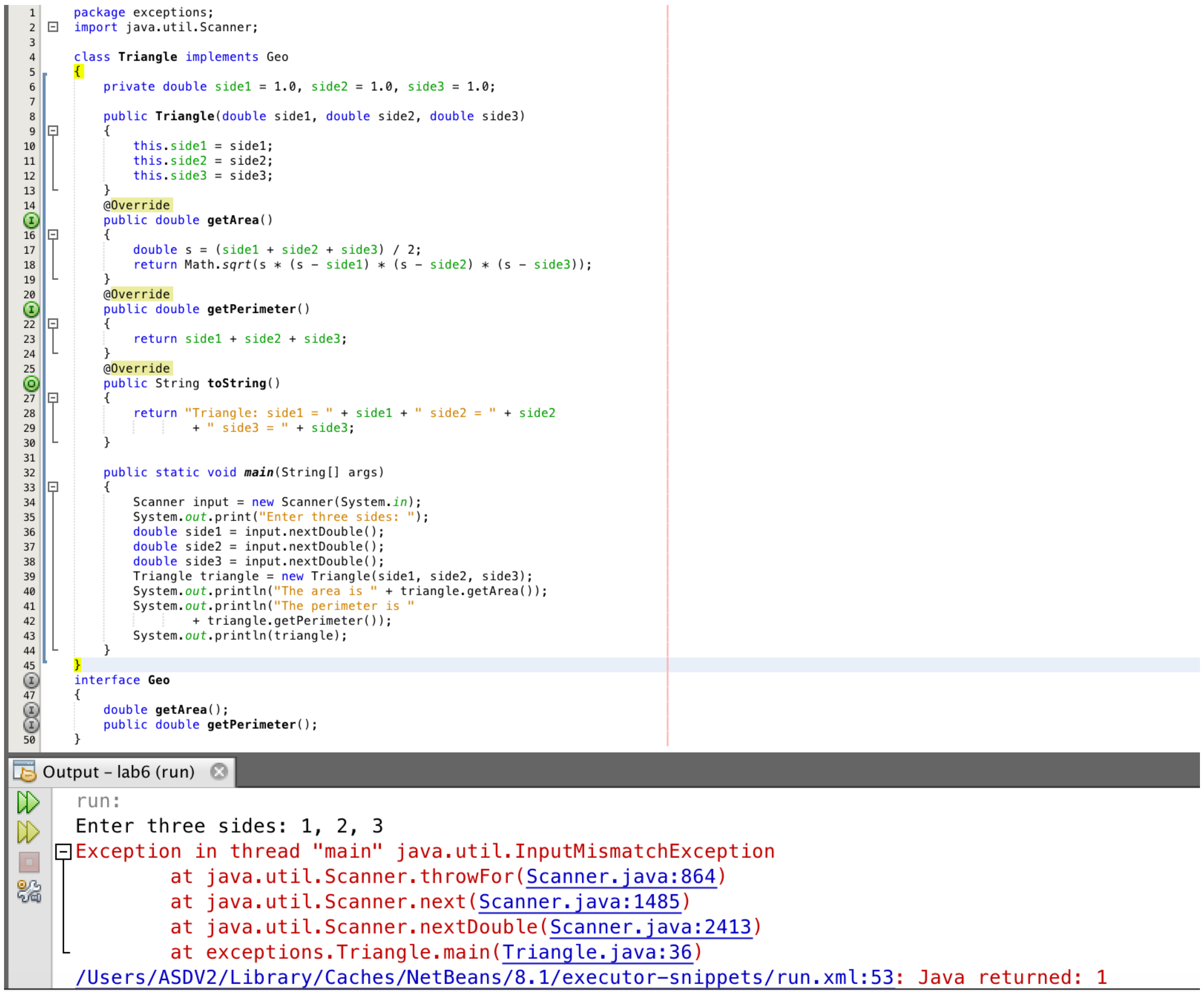


```java
package exceptions;
public class CircleWithUncheckedException
{
    private double radius;

    public CircleWithUncheckedException(){ this( 1.0);}
    public CircleWithUncheckedException(double radius)
    {
        if ( radius < 0 )
            throw new IllegalArgumentException( "radius cannot be negative");
        this.radius = radius;
    }
    public double getRadius()
    {
        return radius;
    }
    public void setRadius(double radius)
    {
        if ( radius < 0 )
            throw new IllegalArgumentException( "radius cannot be negative");
        this.radius = radius;
    }

    @Override
    public String toString()
    {
        return "CircleWithUncheckedException{" + "radius=" + radius + '}';
    }

 public static void main(String...args)
 {
     try
     {
         System.out.println( new CircleWithUncheckedException(5) );
         System.out.println( new CircleWithUncheckedException(-5) );
         System.out.println( new CircleWithUncheckedException(10) );
     }
     catch( Exception e)
     {
         System.err.println( e.getMessage());
     }
 }
}
```

Output – lab6 (run)

```
run:
radius cannot be negative
CircleWithUncheckedException{radius=5.0}
BUILD SUCCESSFUL (total time: 0 seconds)
```

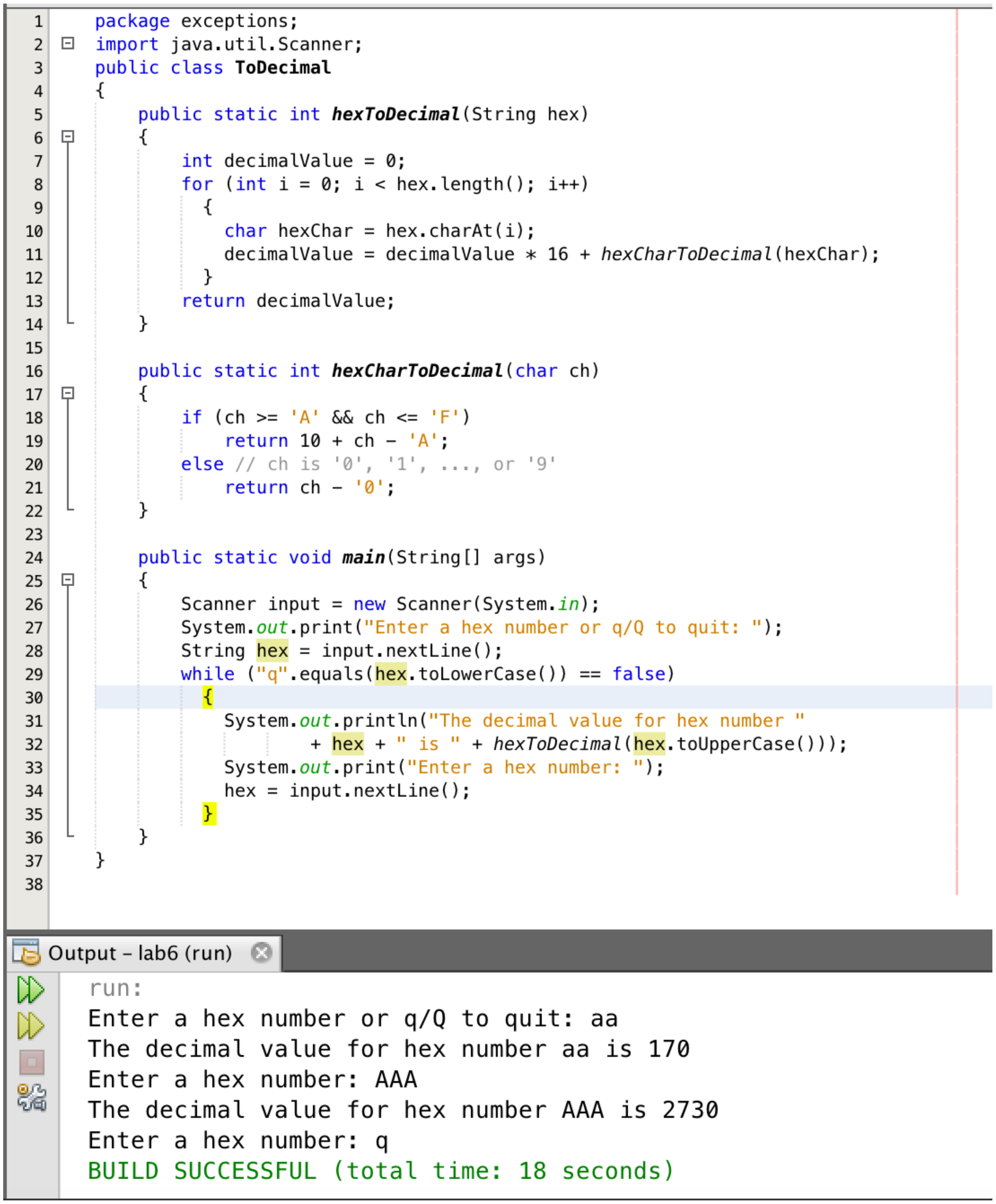
3. Implement the class Triangle shown and run it for the input shown to generate the exception shown. In a triangle, the sum of any two sides is greater than the other side. The Triangle class must adhere to this rule. Create the IllegalTriangleException class, and modify the constructor of the Triangle class to throw an IllegalTriangleException object if a triangle is created with sides that violate the rule, as follows:
public Triangle(double side1, double side2, double side3) throws IllegalTriangleException { // Implement it }



```
1   package exceptions;
2   import java.util.Scanner;
3
4   class Triangle implements Geo
5   {
6       private double side1 = 1.0, side2 = 1.0, side3 = 1.0;
7
8       public Triangle(double side1, double side2, double side3)
9       {
10          this.side1 = side1;
11          this.side2 = side2;
12          this.side3 = side3;
13      }
14      @Override
        public double getArea()
16      {
17          double s = (side1 + side2 + side3) / 2;
18          return Math.sqrt(s * (s - side1) * (s - side2) * (s - side3));
19      }
20      @Override
        public double getPerimeter()
22      {
23          return side1 + side2 + side3;
24      }
25      @Override
        public String toString()
27      {
28          return "Triangle: side1 = " + side1 + " side2 = " + side2
29                  + " side3 = " + side3;
30      }
31
32      public static void main(String[] args)
33      {
34          Scanner input = new Scanner(System.in);
35          System.out.print("Enter three sides: ");
36          double side1 = input.nextDouble();
37          double side2 = input.nextDouble();
38          double side3 = input.nextDouble();
39          Triangle triangle = new Triangle(side1, side2, side3);
40          System.out.println("The area is " + triangle.getArea());
41          System.out.println("The perimeter is "
42                  + triangle.getPerimeter());
43          System.out.println(triangle);
44      }
45  }
    interface Geo
47  {
        double getArea();
        public double getPerimeter();
50  }
```

```
run:
Enter three sides: 1, 2, 3
Exception in thread "main" java.util.InputMismatchException
	at java.util.Scanner.throwFor(Scanner.java:864)
	at java.util.Scanner.next(Scanner.java:1485)
	at java.util.Scanner.nextDouble(Scanner.java:2413)
	at exceptions.Triangle.main(Triangle.java:36)
/Users/ASDV2/Library/Caches/NetBeans/8.1/executor-snippets/run.xml:53: Java returned: 1
```

4. The hex2Dec(String hexString) method, which converts a hex string into a decimal number. Re-implement the hex2Dec method to throw a NumberFormatException if the string is not a hex string. Catch the exception inside the while loop of main and stay in the program until the user types q/Q.



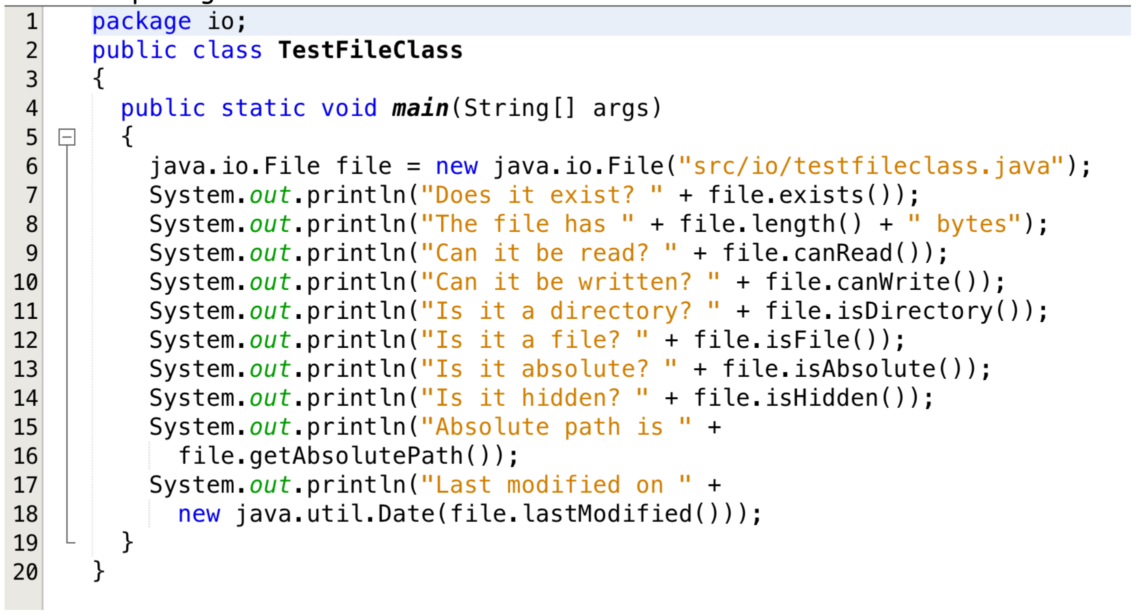
```java
package exceptions;
import java.util.Scanner;
public class ToDecimal
{
    public static int hexToDecimal(String hex)
    {
        int decimalValue = 0;
        for (int i = 0; i < hex.length(); i++)
          {
            char hexChar = hex.charAt(i);
            decimalValue = decimalValue * 16 + hexCharToDecimal(hexChar);
          }
        return decimalValue;
    }

    public static int hexCharToDecimal(char ch)
    {
        if (ch >= 'A' && ch <= 'F')
            return 10 + ch - 'A';
        else // ch is '0', '1', ..., or '9'
            return ch - '0';
    }

    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter a hex number or q/Q to quit: ");
        String hex = input.nextLine();
        while ("q".equals(hex.toLowerCase()) == false)
          {
            System.out.println("The decimal value for hex number "
                    + hex + " is " + hexToDecimal(hex.toUpperCase()));
            System.out.print("Enter a hex number: ");
            hex = input.nextLine();
          }
    }
}
```

Output - lab6 (run)

```
run:
Enter a hex number or q/Q to quit: aa
The decimal value for hex number aa is 170
Enter a hex number: AAA
The decimal value for hex number AAA is 2730
Enter a hex number: q
BUILD SUCCESSFUL (total time: 18 seconds)
```

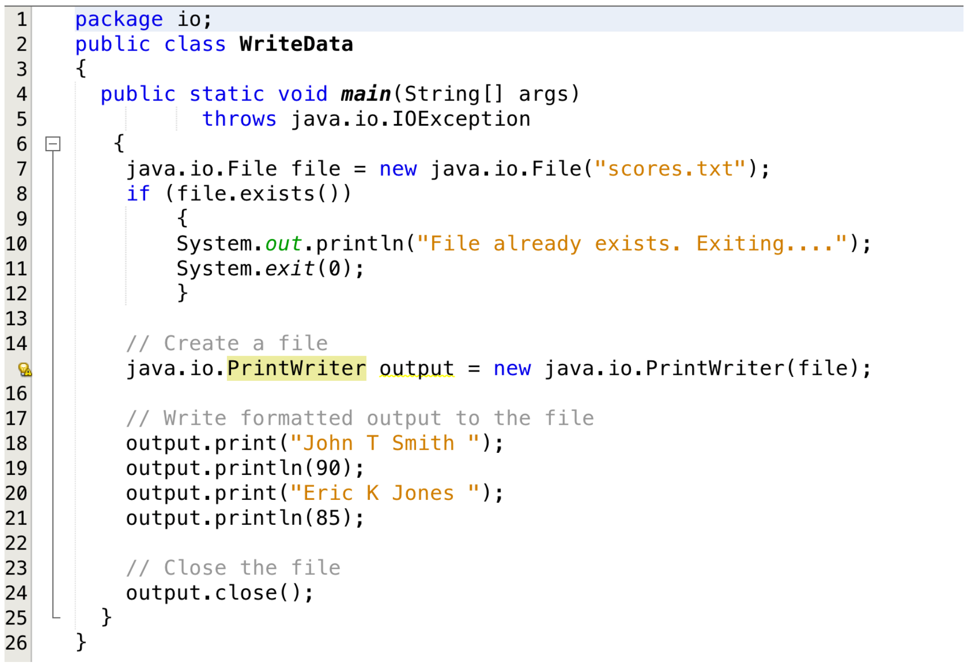
5. Add method the <u>bin2Dec(String binaryString)</u> to previous class <u>ToDecimal</u> to convert a binary string into a decimal number. Implement the bin2Dec method to throw a NumberFormatException if the string is not a binary string. Test it with code from main as follows: The user has options q/Q, b/B, x/X to quit, or enter a binary or enter a hex.

6. Add to your existing project (any) a new package called ***io***.
   Under package ***io*** add the class shown below. Run it. Test the code for different files.



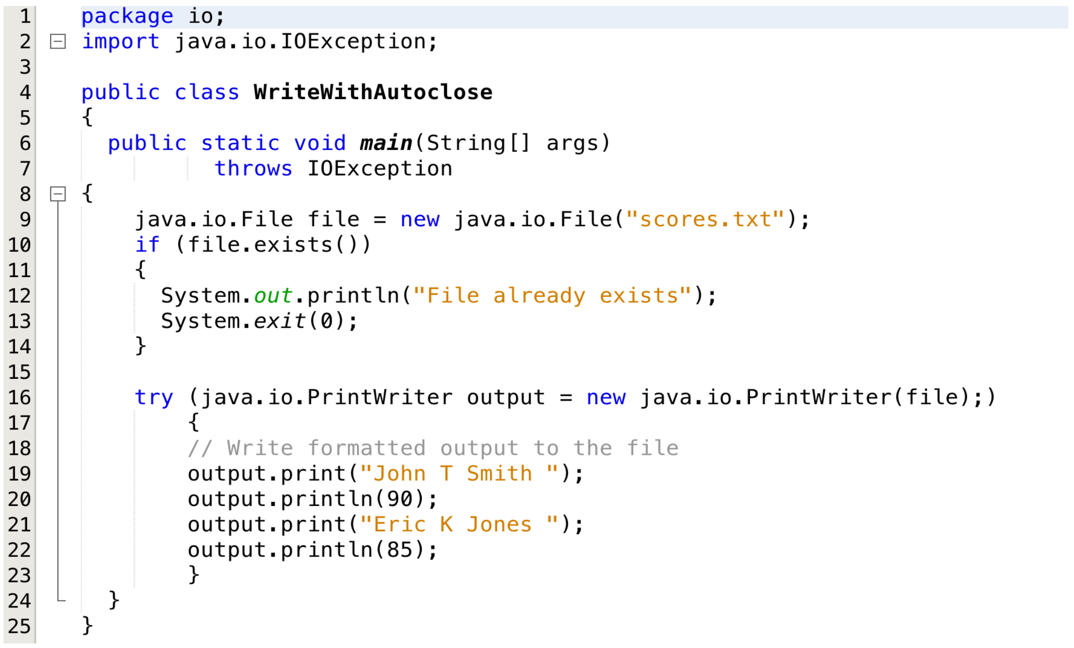
```java
package io;
public class TestFileClass
{
  public static void main(String[] args)
  {
    java.io.File file = new java.io.File("src/io/testfileclass.java");
    System.out.println("Does it exist? " + file.exists());
    System.out.println("The file has " + file.length() + " bytes");
    System.out.println("Can it be read? " + file.canRead());
    System.out.println("Can it be written? " + file.canWrite());
    System.out.println("Is it a directory? " + file.isDirectory());
    System.out.println("Is it a file? " + file.isFile());
    System.out.println("Is it absolute? " + file.isAbsolute());
    System.out.println("Is it hidden? " + file.isHidden());
    System.out.println("Absolute path is " +
      file.getAbsolutePath());
    System.out.println("Last modified on " +
      new java.util.Date(file.lastModified()));
  }
}
```

7. Implement the class below:



```java
package io;
public class WriteData
{
  public static void main(String[] args)
        throws java.io.IOException
   {
    java.io.File file = new java.io.File("scores.txt");
    if (file.exists())
        {
        System.out.println("File already exists. Exiting....");
        System.exit(0);
        }

    // Create a file
    java.io.PrintWriter output = new java.io.PrintWriter(file);

    // Write formatted output to the file
    output.print("John T Smith ");
    output.println(90);
    output.print("Eric K Jones ");
    output.println(85);

    // Close the file
    output.close();
  }
}
```

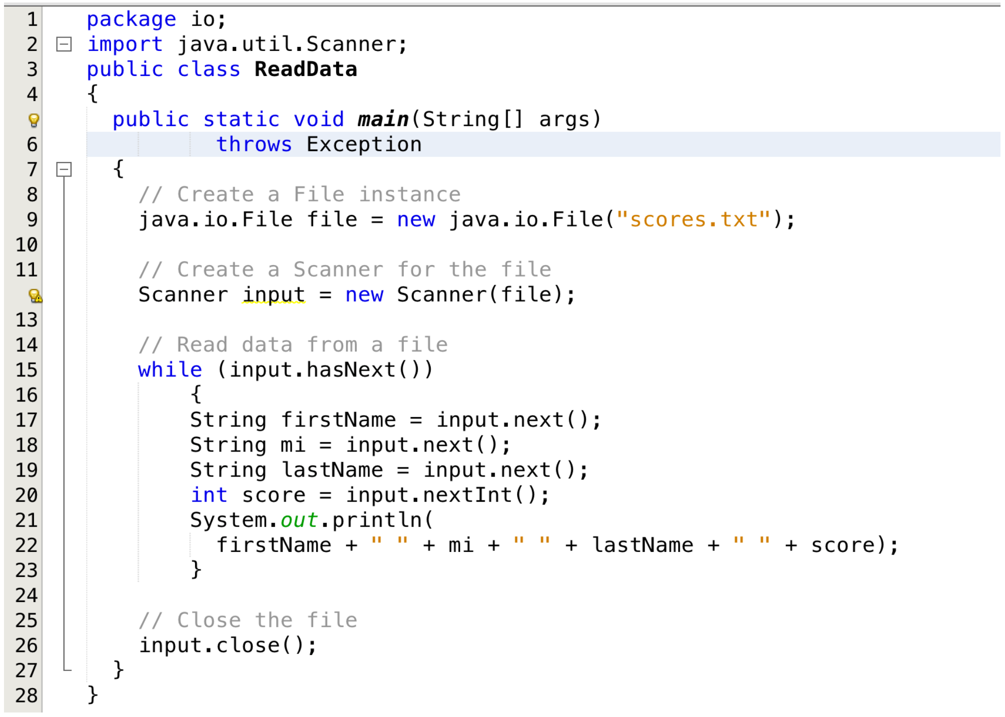
8. Implement the class below:



```
package io;
import java.io.IOException;

public class WriteWithAutoclose
{
  public static void main(String[] args)
          throws IOException
{
    java.io.File file = new java.io.File("scores.txt");
    if (file.exists())
    {
      System.out.println("File already exists");
      System.exit(0);
    }

    try (java.io.PrintWriter output = new java.io.PrintWriter(file);)
        {
        // Write formatted output to the file
        output.print("John T Smith ");
        output.println(90);
        output.print("Eric K Jones ");
        output.println(85);
        }
  }
}
```

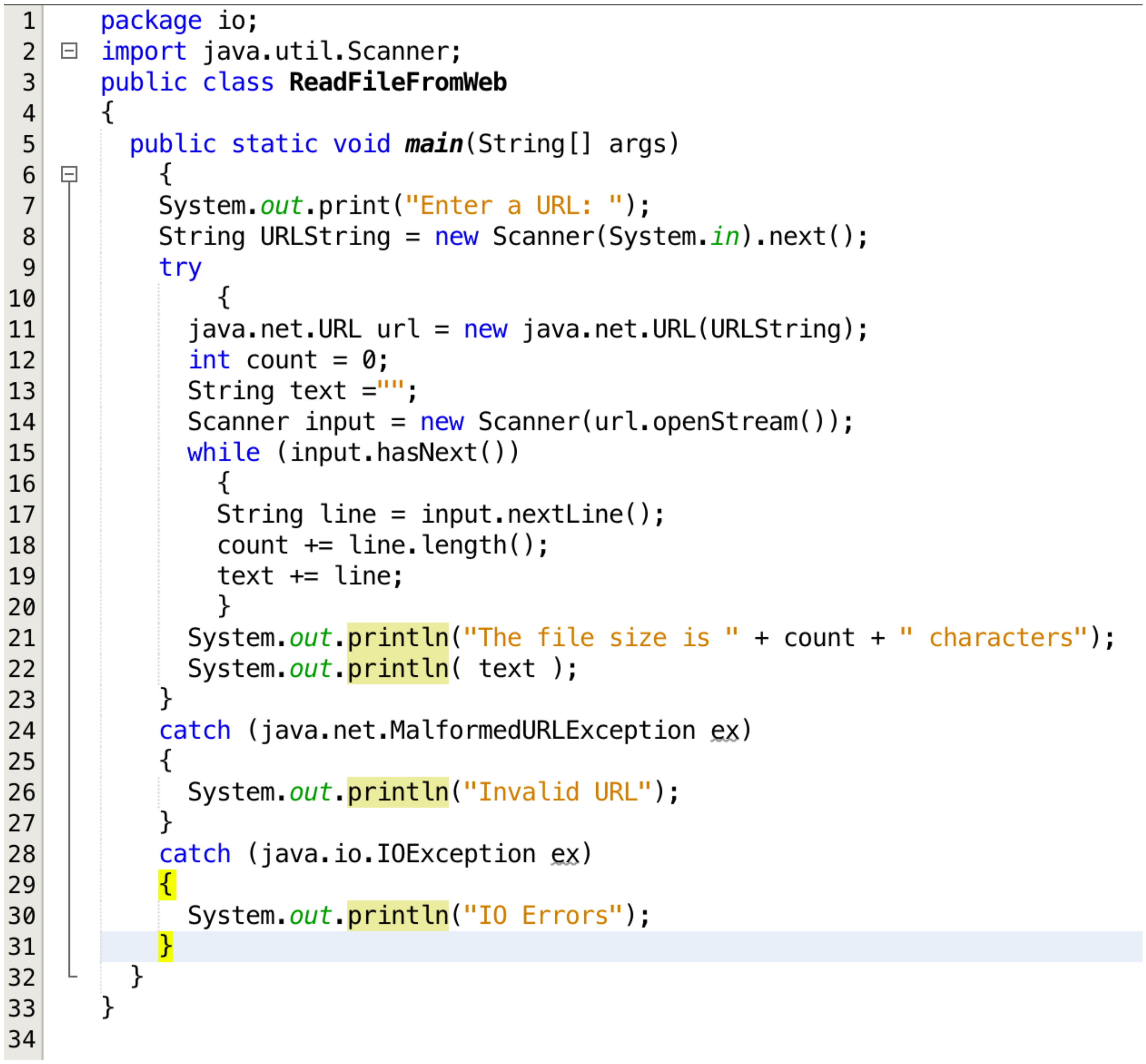
9. Add a new class <u>WriteWithAutoClose2</u>. The new class modifies the *WriteWithAutoClose:* The new class doesn't declare that throws an exception but handles the exception locally and display a message inside the catch block ---msg: "IOException caught"

10. Implement the class below:



```
package io;
import java.util.Scanner;
public class ReadData
{
  public static void main(String[] args)
          throws Exception
  {
    // Create a File instance
    java.io.File file = new java.io.File("scores.txt");

    // Create a Scanner for the file
    Scanner input = new Scanner(file);

    // Read data from a file
    while (input.hasNext())
        {
        String firstName = input.next();
        String mi = input.next();
        String lastName = input.next();
        int score = input.nextInt();
        System.out.println(
          firstName + " " + mi + " " + lastName + " " + score);
        }

    // Close the file
    input.close();
  }
}
```

11. Implement the class below. Test it with full-paths of webpages that end with the html extension.



```
package io;
import java.util.Scanner;
public class ReadFileFromWeb
{
 public static void main(String[] args)
  {
  System.out.print("Enter a URL: ");
  String URLString = new Scanner(System.in).next();
  try
    {
   java.net.URL url = new java.net.URL(URLString);
   int count = 0;
   String text ="";
   Scanner input = new Scanner(url.openStream());
   while (input.hasNext())
    {
    String line = input.nextLine();
    count += line.length();
    text += line;
    }
   System.out.println("The file size is " + count + " characters");
   System.out.println( text );
  }
  catch (java.net.MalformedURLException ex)
  {
   System.out.println("Invalid URL");
  }
  catch (java.io.IOException ex)
  {
   System.out.println("IO Errors");
  }
 }
}
```

12. Modify ReadFileWebPage to count all <div> and all <p>. Display their numbers. The new class is called *ReadFileWebPage1.*

13. *Create class* ProcessScoresInTextFile. *Suppose that a text file contains an unspecified* number of scores separated by blanks. The class prompts the user to enter the file, reads the scores from the file, and displays their total and average. Scores are separated by blanks.

14. *Create class* WriteReadData . The class creates a file named rw.txt if it does not exist. Writes 100 integers created randomly into the file using text I/O. Integers are separated by spaces in the file. Read the data back from the file and display the sorted data in your screen.

15. Modify the file below so when you type an invalid URL it stays in a loop until the user types a valid URL.

```java
import java.util.Scanner;

public class ReadFileFromURL {
  public static void main(String[] args) {
    System.out.print("Enter a URL: ");
    String URLString = new Scanner(System.in).next();

    try {
      java.net.URL url = new java.net.URL(URLString);
      int count = 0;
      Scanner input = new Scanner(url.openStream());
      while (input.hasNext()) {
        String line = input.nextLine();
        count += line.length();
      }

      System.out.println("The file size is " + count + " characters");
    }
    catch (java.net.MalformedURLException ex) {
      System.out.println("Invalid URL");
    }
    catch (java.io.IOException ex) {
      System.out.println("IO Errors");
    }
  }
}
```

16.