CIS 36A– 24th In Class / Lab Assignment – **10 Points**

Student Name____________________________ **Student ID**____________________________ **Point Total**

http://beginnersbook.com/java-tutorial-for-beginners-with-examples/

http://beginnersbook.com/2013/04/java-exception-handling/

Pg 347 – 355 Java Programming *A comprehensive Introduction*

Section 1: Define

Describe the difference between an **Error** in Java and an **Exception.**

Errors should not be caught or handled (except in the rarest of cases). Exceptions are the bread and butter of exception handling.

Describe the difference between **Checked Exceptions** and **Unchecked Exceptions**

Checked: are the exceptions that are checked at compile time. If some code within a method throws a checked exception, then the method must either handle the exception or it must specify the exception using throws keyword.

Unchecked are the exceptions that are not checked at compiled time.

Give a few examples of **Checked Exceptions.**

```
class Main {
    public static void main(String[] args) {
        FileReader file = new FileReader(”C:¥¥test¥¥a.txt”);
        BufferedReader fileInput = new BufferedReader(file);

        // Print first 3 lines of file ”C:¥test¥a.txt”
        for (int counter = 0; counter < 3; counter++)
            System.out.println(fileInput.readLine());

        fileInput.close();
    }
}
```

Output:

```
Exception in thread "main" java.lang.RuntimeException: Uncompilable source code -
unreported exception java.io.FileNotFoundException; must be caught or declared to
be
thrown
	at Main.main(Main.java:5)
```

Give a few examples of **Unchecked Exceptions.**

```
class Main {
   public static void main(String args[]) {
      int x = 0;
      int y = 10;
      int z = y/x;
  }
}
```

Output:

```
Exception in thread "main" java.lang.ArithmeticException: / by zero
	at Main.main(Main.java:5)
Java Result: 1
```

Describe basic structure of

try{

}

Catch

{

}

A try statement is used to catch exceptions that might be thrown as your program executes. You should use a try statement whenever you use a statement that might throw an exception That way, your program won’t crash if the exception occurs.

CIS 36A– 24th In Class / Lab Assignment – **10 Points**

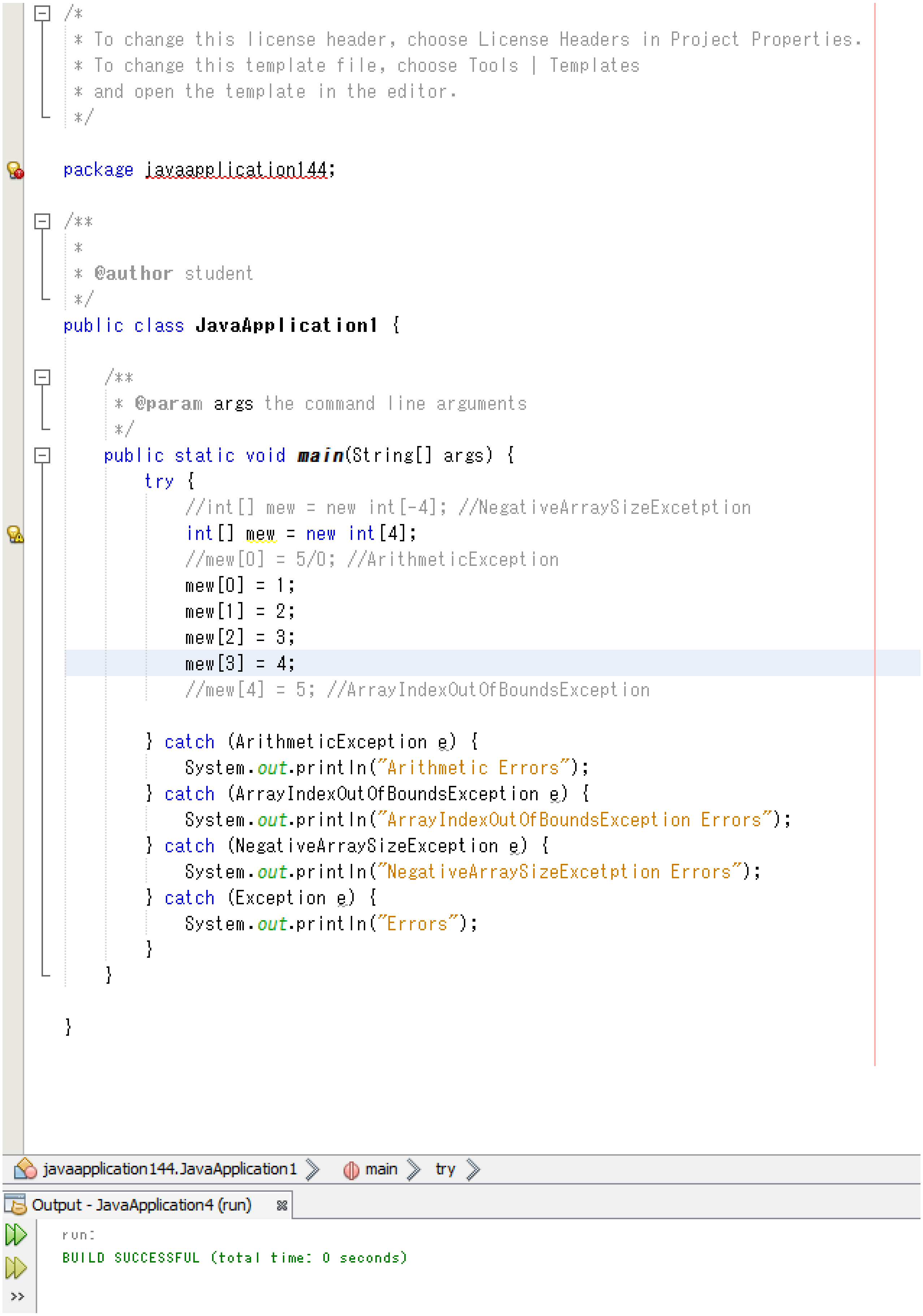
Student Name_______________________________ **Student ID**__________________________________ **Point Total**

Programming Tasks:

Task 1- Create coded examples of ArithmeticException, ArrayIndexOutOfBoundsException, and NegativeArraySizeException Handling. Do not use the exact example from the book. Create your own.

CIS 36A– 24th   In Class / Lab Assignment – **10 Points**

Student Name______________________ **Student ID**______________________ **Point Total**



```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package javaapplication144;

/**
 *
 * @author student
 */
public class JavaApplication1 {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        try {
            //int[] mew = new int[-4]; //NegativeArraySizeExcetption
            int[] mew = new int[4];
            //mew[0] = 5/0; //ArithmeticException
            mew[0] = 1;
            mew[1] = 2;
            mew[2] = 3;
            mew[3] = 4;
            //mew[4] = 5; //ArrayIndexOutOfBoundsException

        } catch (ArithmeticException e) {
            System.out.println("Arithmetic Errors");
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("ArrayIndexOutOfBoundsException Errors");
        } catch (NegativeArraySizeException e) {
            System.out.println("NegativeArraySizeExcetption Errors");
        } catch (Exception e) {
            System.out.println("Errors");
        }
    }

}
```

javaapplication144.JavaApplication1
main
try
Output - JavaApplication4 (run)

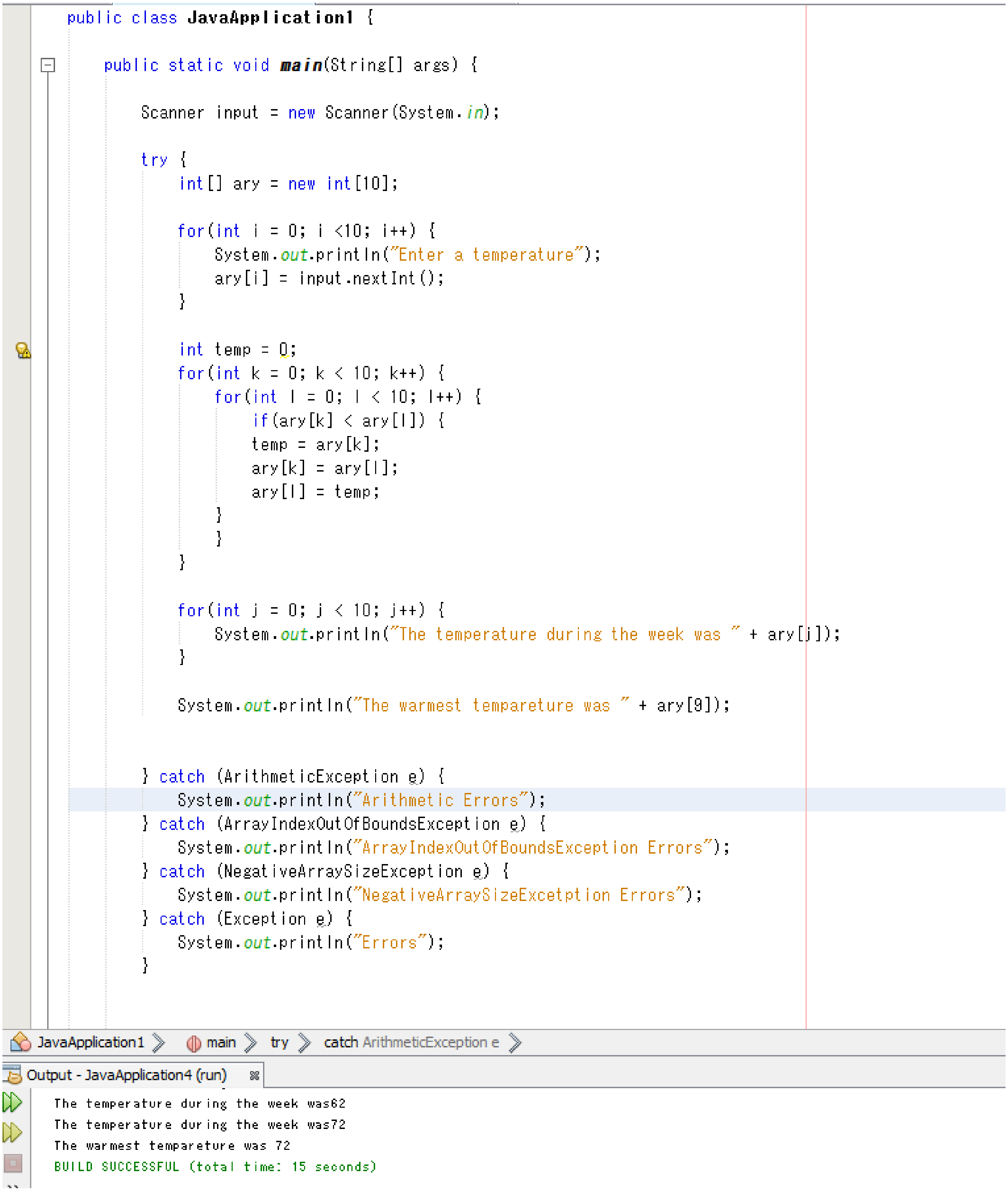
```
run:
BUILD SUCCESSFUL (total time: 0 seconds)
```

CIS 36A– 24th In Class / Lab Assignment – **10 Points**
Student Name_____________________________ **Student ID**_____________________________ **Point Total**

Task 2- Incorporate exception handling into anyone of the previous programming tasks have completed. Make it clear the change in code.

CIS 36A– 24th   In Class / Lab Assignment – **10 Points**

Student Name________________________________ **Student ID**______________________________________ **Point Total**



```
public class JavaApplication1 {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        try {
            int[] ary = new int[10];

            for(int i = 0; i <10; i++) {
                System.out.println("Enter a temperature");
                ary[i] = input.nextInt();
            }

            int temp = 0;
            for(int k = 0; k < 10; k++) {
                for(int l = 0; l < 10; l++) {
                    if(ary[k] < ary[l]) {
                    temp = ary[k];
                    ary[k] = ary[l];
                    ary[l] = temp;
                }
                }
            }

            for(int j = 0; j < 10; j++) {
                System.out.println("The temperature during the week was " + ary[j]);
            }

            System.out.println("The warmest tempareture was " + ary[9]);


        } catch (ArithmeticException e) {
            System.out.println("Arithmetic Errors");
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("ArrayIndexOutOfBoundsException Errors");
        } catch (NegativeArraySizeException e) {
            System.out.println("NegativeArraySizeExcetption Errors");
        } catch (Exception e) {
            System.out.println("Errors");
        }
```

JavaApplication1 > main > try > catch ArithmeticException e >
Output - JavaApplication4 (run)

```
The temperature during the week was62
The temperature during the week was72
The warmest tempareture was 72
BUILD SUCCESSFUL (total time: 15 seconds)
```