Tutorial 6 Exceptions

1. Having seen an example of how to create an arithmetic exception in the lecture slides i.e. System.out.println(number1/number2); when variable number2 is 0, create a new class named Test with a main method. Write code in the main method that will cause the Java default exception handler to output the following stack trace. (Your output doesn't have to have the exact line number and exception message 10).

```
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 10
at Test1.main(Test1.java:7)
```

2. Write a try-block and catch-block to handle the exception. Once the exception is caught print the string of the exception and exception message on separate lines.

3. If we want to catch all exceptions some block of code will throw. What should we put in the () at the start of the catch blocks?

4. Why does this work?

5. In most situations, we want our exception handler to be as specific as possible i.e. being able to identify the exact type of exception being thrown rather than identifying exceptions by the most generic Exception class. Why is that?

6. We can also arbitrarily throw exceptions in our code. Inside the main method, write code to arbitrarily throw a NoSuchMethodException inside a try-block followed by a catch-block that catches the most generic exception.

7. It is also possible to write a customised exception. Write a customised exception named [your name]sSpecialException. In your customised exception, overwrite the getMessage() and getLocalizedMessage with your own messages. Also implement a method getOrigin() that returns a string “This exception was made in a Compsci 230 lab”.

8. Throw an instance of your customized exception inside a try-block. Catch it using a specific catch block and print out the result of the getMessage(), getLocalizedMessage() and getOrigin() methods.