**COMP 132 - Homework # 9**

**Exceptions**

1. Consider the following class definitions:

**public** **class** WhatsTheOutput {

**public** **static** **void** main(String[] args) {

System.*out*.println("Main Starting");

**try** {

*methodOne*();

}

**catch** (ExceptionTypeE e) {

System.*out*.println("Caught type E");

}

System.*out*.println("Main Ending");

}

**public** **static** **void** methodOne() {

System.*out*.println("MethodOne Starting");

**try** {

*methodTwo*();

}

**catch** (ExceptionTypeC e) {

System.*out*.println("Caught type C");

}

System.*out*.println("MethodOne Ending");

}

**public** **static** **void** methodTwo() {

System.*out*.println("MethodTwo Starting");

**try** {

**// THROW EXCEPTION HERE!**

}

**catch** (ExceptionTypeA e) {

System.*out*.println("Caught type A");

}

System.*out*.println("MethodTwo Ending");

}

}

**public class** ExceptionTypeA **extends** RuntimeException {…}

**public class** ExceptionTypeB **extends** RuntimeException {…}

**public class** ExceptionTypeC **extends** ExceptionTypeB {…}

**public class** ExceptionTypeD **extends** ExceptionTypeC {…}

**public class** ExceptionTypeE **extends** RuntimeException {…}

Indicate the output that the above program would produce if the following exception types were thrown on the line labeled “THROW EXCEPTION HERE”. If the program would crash also indicate that by writing “CRASH!” at the end of the output.

 a. ExceptionTypeA

 b. ExceptionTypeB

 c. ExceptionTypeC

 d. ExceptionTypeD

 e. ExceptionTypeE

2. Rewrite methodOne from question #1 such that it catches exceptions for type ExceptionTypeD and ExceptionTypeE and prints distinct messages in each case. Give just your modified code for methodOne.

3. Many methods in the Java Development Kit (JDK) throw exceptions when they are unable to perform the requested action. The JavaDoc for each method describes the type of exceptions that the method might throw and the circumstances under which they are thrown. Use the JDK documentation (linked to from the course home page) to identify the types of exceptions that might be thrown by the methods below. Also indicate if each exception type is **Checked** or **Unchecked**.

 a. Double.parseDouble(String val)

 b. ArrayList.get(int index)

 c. Thread.sleep(long millis, int nanos)

 d. File.getCanonicalPath()

 e. JTextField.setColumns(int columns)

4. The following questions use the PhoneNumber example from the source code available on the course homepage.

a. Try to create a new PhoneNumber with an invalid number (e.g. “(abc) 245-1401”). What type of exception do you receive? In which method in the PhoneNumber class does the exception occur? On which line of the PhoneNumber class is the exception generated?

b. Try to create a new PhoneNumber that does not contain enough digits (e.g. “(717) 245-140”). What type of exception do you receive? In which method in the PhoneNumber class does the exception occur? On which line of the PhoneNumber class is the exception generated?

c. Modify the PhoneNumber constructor so that it throws an IllegalArgumentException, with a descriptive message, instead of the different exception types identified in parts a and b. Paste your constructor code as your solution to this question.

d. Write a snippet of code that reads a phone number from the user. If an invalid phone number is entered your code should print an error message and the user should be prompted again. Paste your code as your solution to this exercise.

e. Write a JUnit test that checks that the PhoneNumber constructor now throws the correct exception when an invalid phone number is provided. Give the code for your JUnit test as the answer to this question.