**Core Java 8-2: Exceptions**

1. ***(Exceptional Conditions)*** List the various exceptional conditions that you have seen so far such as ArithmeticException, ArrayIndexOutOfBoundsException, NullPointerException, NoSuchElementException or IllegalStateException (Scanner) or others. List as many additional exceptional conditions as you can and just describe them from the viewpoint of where you may have or will see them and what you can do to handle the exceptions.
2. ***(Exceptions and Constructor Failure)*** Can exception handling be used with constructors and with problems with constructors? Please explain.
3. ***(Catching Exceptions with Superclasses)*** Use inheritance to create an exception superclass (called ExceptionA) and exception subclasses ExceptionB and ExceptionC, where ExceptionB inherits from ExceptionA and ExceptionC inherits from ExceptionB. **Write a program to demonstrate that the catch block for type ExceptionA catches exceptions of types ExceptionB and E**xceptionC. These classes will be custom exceptions.
4. ***(Catching Exceptions Using Class***Exception***)*** Write a program that demonstrates how various exceptions are caught with

catch (Exception exception)

This time, define classes ExceptionA (which inherits from class Exception) and ExceptionB (which inherits from class ExceptionA). In your program, create try blocks that throw exceptions of types ExceptionA, ExceptionB, NullPointerException and IOException. All exceptions should be caught with catch blocks specifying type Exception.

1. ***(Order of***catch***Blocks)*** Write a program demonstrating that the order of catch blocks is important. If you try to catch a superclass exception type before a subclass type, the compiler should generate errors. Why do you think this is the case?
2. ***(Constructor Failure)*** Write a program that shows a constructor passing information about constructor failure to an exception handler. Define class SomeClass, which throws an Exception in the constructor. Your program should try to create an object of type SomeClass and catch the exception that’s thrown from the constructor.
3. ***(Rethrowing Exceptions)*** Write a program that illustrates rethrowing an exception. Define methods someMethod and someMethod2. Method someMethod2 should initially throw an exception. Method someMethod should call someMethod2, catch the exception and rethrow it. Call someMethod from method main, and catch the rethrown exception. Print the stack trace of this exception. Is it possible or a good idea to catch an exception in a catch block and throw a different type of exception? Please describe. Note: Go to the JavaDocs and check the Exception class constructor that throws an has an argument Throwable cause or the overloaded constructors with this parameter.
4. ***(Catching Exceptions Using Outer Scopes)***