**Week 5 Research Project**

Timothy King

**5.) What is an Exception?**

<https://search.brave.com/search?q=In+Jave%2C+explain+what+an+Exception+is.&source=web&summary=1&conversation=e90c4bd8bd7b44858c31e6>

[https://search.brave.com/search?q=In+Java%2C+concerning+Exceptions%2C+is+it+possible+for+Exceptions+to+be+caused+intentionally%2C+by+users%2C+to+break%2C+or+break+%28hack%29+into+a+program+%2F+computer%2C+so+as+to+do+something+malicious%2C+such+as+steal+o](https://search.brave.com/search?q=In+Java%2C+concerning+Exceptions%2C+is+it+possible+for+Exceptions+to+be+caused+intentionally%2C+by+users%2C+to+break%2C+or+break+(hack)+into+a+program+%2F+computer%2C+so+as+to+do+something+malicious%2C+such+as+steal+or+corrupt+data%2C+etc%3F&source=web&summary=1&conversation=56ec1e0d800942f433f831)

Exceptions are errors or events that are unexpected, caused either by coding errors, user input errors, or generally unpredictable events, such as the network going down. The system creates an object with the exception in it, with the intent that someone will use it. This can be caught and handled, perhaps writing it to a log file, send an alert to the user, and/or allow some action to be taken that will prevent the program from crashing. Proper exception handling is important, because it allows the developer to correct problems with their code.

It is also possible for users to intentionally cause exceptions, such as through intentionally entering input that the program is not expecting, so as to allow access to a system command prompt, inject arbitrary code, corrupt system files, or steal data, etc. This escalates the importance of proper exception handling from merely a "programming best practice," to an existential corporate security essential, typically referred to as "DevSecOps." The idea is that you must consider Security from the very beginning of designing and writing your code. Within that, programmers must ensure that all exceptions are caught and properly handled, all input is validated and restricted to the intended data types and lengths, etc, and passwords and other sensitive information is encrypted within the program, as well as during storage and transmission. And lastly, supervisors and testers must specifically look for those things and verify that they exist and have been done properly, prior to the code being released.

**6.) What are the differences between Checked and Unchecked Exceptions?**

<https://search.brave.com/search?q=In+Java%2C+what+is+the+difference+between+checked+and+unchecked+exceptions%2C+and+how+do+you+handle+them%3F&source=web&summary=1&conversation=01a1637bf9d9619dffeb3a>

The primary difference between checked and unchecked exceptions, is that checked exceptions are looked for by the compiler, at compile time, and they consist of predictable errors that the program should be able to recover from, while unchecked exceptions do not get checked at compile time by the compiler, because they are unpredictable and are caused by environmental conditions, user input errors, and so on. Also, unchecked errors come from the RuntimeException class, such as NullPointerExceptions, OutOfBoundsExceptions, etc; whereas, checked errors come from the rest of the Exception class, such as FileNotFoundExceptions, IOExceptions, etc. Both types of exceptions can be handled with a try-catch block, or they can be thrown to a higher level of code with a throw statement, as part of a method declaration, for instance, when they can be better handled at the higher level.