Davis Klevgard

Java Web Programming

**Exceptions**

What is an Exception? An exception is what happens when there is a problem that may happen within your code. This sort of problem usually happens unexpectedly because the user of the program may input data or data may not exist where you, the programmer, do not expect someone to put in the wrong data for your code. It has been said, that if a user can think of a way to break your code, they will. So, an Exception will be something that a programmer will try to catch within their program and come up with a solution if the problem arises. These problems have different ways of showing up. One is compile-time, which is considered a checked exception, and the second is during run-time, with that being considered an unchecked exception. These problems can also happen a lot of times and mostly during the development of many programs.

So, what are the differences between Checked and Unchecked Exceptions? Well as previously stated before; Checked occurs during the compile-time of a program, while unchecked happens during the run-time. Compile time is when the program is being converted to machine code, while run time is when the program is running. Checked exceptions must be handled in the method that has that exception, either by fixing the problem or using throws to get rid of the exception. Throws can be used for both checked and unchecked exceptions though. Unchecked exceptions though can be solved during the run time, so methods don’t need to throw those exceptions right away.

So why do we even need to make sure we handle these exceptions? Well, we programmers must figure out how to handle these problems so that the program does not cause the clients we are building these programs for, to lose money, data, their own clients, or dreams that they put their lives and futures into. These things may happen because the program that we provided for them have issues within the code that may cause repeated problems. That will also affect our livelihood for the fact that we would be labeled as providers of bad code.

With all these things to consider, I will need a plan to provide code that does not crash or not run properly for my clients. First thing to understand is what problems my code may have when it goes into production. This will entail keeping a detailed account of what events will produce exceptions and errors. The second part is learning how to fix those problems so that they do not become an issue. That will mean learning on my own and working to understand how to fix the problems. With this plan in place, I should be able to learn and understand how to reduce exceptions within my code.

**Resources**

"Java - Exceptions". *Tutorialspoint.Com*, 2021, <https://www.tutorialspoint.com/java/java_exceptions.htm>.

"Exceptions In Java - Geeksforgeeks". *Geeksforgeeks*, 2021, <https://www.geeksforgeeks.org/exceptions-in-java/>.

"Checked Vs Unchecked Exceptions In Java - Geeksforgeeks". *Geeksforgeeks*, 2021, <https://www.geeksforgeeks.org/checked-vs-unchecked-exceptions-in-java/>.

"Java Exceptions Hierarchy Explained". *Rollbar*, 2021, <https://rollbar.com/blog/java-exceptions-hierarchy-explained/>.

"How To Handle Exceptions In Java". *Rollbar*, 2021, <https://rollbar.com/guides/java/how-to-handle-exceptions-in-java/>.