In Java there are two types of exceptions. The first being checked exceptions and the second being unchecked exceptions. With checked exceptions these are checked at compile time. When we have unchecked exceptions, these are not checked at compile time and the complier is not forced to handle the exception. “It is up to the programmers to be civilized, and specify or catch the exceptions. In Java exceptions under Error and RuntimeException classes are unchecked exceptions, everything else under throwable is checked.” (geeksforgeeks.org) As a whole there seems to be a debate as to what is best to be used. However, from what I have read the overall principals should be; when a program is designed and there can be reasonable expectations from a user to recover from an exception, it should be checked. If that is not the case it should be an unchecked exception.

In all programs it is possible to throw errors. Sometimes these errors come from the wrong input from a user or unexpected things. This causes the compiler to throw and error. This is where the try statements can really help programmers. It gives us the ability to try and forsee issues ahead of time and it allows us to specify what should be done in certain conditions. For example if a program is being built that asks the user to input a number form 1-10 and the user types in string, we can “catch” that error and properly program a re-prompt to the user instead of it crashing the program. A simple example could be (taken form w3schools.com):