Jennifer Swofford  
APC 390  
Assignment4: Crash report

# Already Handled Exceptions

When I initially built this program, I handled many possible exceptions using built-in java exceptions. Rather than ‘re-invent the wheel’ with these exceptions I’ve left them as is:

addRow:   
has a catch for ArrayIndexOutOfBoundsException. This gets thrown when a user select add row when they already have a full table. It also would get hit when the user loads a file with over 100 rows.

loadTableFromFile:   
has a catch for NoSuchElementException. This gets thrown when a loaded file is empty.  
has a catch for FileNotFoundException. This gets thrown when a file doesn’t exists.

# Unhandled Exceptions

There are still issues that take place with the string split & the the addRow call in my loadTableFromFile method. This part of my loadTableFromFile method is dependent on the data in the file to be in a specific format. If the user were to select option ‘1’ to load a city file, but the file they try to load is actually a stadium file – the string split and addRow methods will cause some array out of bounds exceptions.   
  
Additionally, my string split expects the loaded table to comma-separate the data with spaces in between them. My current thought process to handle the general file content issues would be to create a custom exception class to advise the user that the file structure needs to be corrected.