The Unit tests I creted combine pretty much all of the token ypes to ensure that when a complex line of code is put into the stream, it is all tokenized correctly. Specifically, the first one involves rreserved keywords, ids , literals , operators and parenthesis combined, the second involves a struct definition, the third involves complex arithmetic and logical expression. These tets are very comprehensive and simulate real code that would be lexed. The negative tests involve unrecognized symbols. Some challenges I faced were incrementing the line because I found it difficult to read a newline without incrementing before actually going to the next line, I addressed this simply by -1 the line in the token constructor before returning. Another challenge was reading multicharacter tokens at first, but it was easy to figure out by using a while loop that reads until a token is recognized.