COSC346 Assignment One Report
Ash Cochrane & Tyler Baker

We are submitting our assignment as an Xcode Project of which we have used Xcodebuild to allow for files to be given as command line arguments. Our submission consists of five .swift files and a single .xcodeproj file, you can also find our testing frameworks separated in the directory "./Releases/Testcases". To run a test case simply move into the ./Release directory and type "./Spreadsheet ./Testcases/filename".

Object-Oriented Concepts:
For the parsing of our code we continued to use the given skeleton code parsing function, however we did proceed to override it majority of the time to perform our calculations. This parsing code is split between two of our files SpreadsheetGrammar.swift and Tokens.swift and both of these files contain a number of classes, each one representing a spreadsheet expression specified to us in the assignment specs. All of our grammar rule classes inherit the GrammarRule class allowing for minimal duplication of code and the overriding of one of its base functions – parse (input: String), this polymorphism lets us use the same function name, the same arguments but a different implementation in each different subclass – something implement widely throughout our code. We also use encapsulated data and associated methods in order to control how the internal state of each object is access and who is allowed to access it.

Testing:
In order to test our code, we set up an initial configuration of the Travis-Cl and then linked it up to our GitHub repository. This allowed us to bring continuous integration into our project as each time we made a commit and pushed to our repository, Travis would try and build the project and run the tests comprehensively. This also allowed us to keep our testing framework separate from our library. To go along with Travis-Cl we implemented several test cases all of which we have submitted along with our swift library.

Pair:
As we completed this assignment in a pair we each divided up equally the classes to complete.

Extensions:
As for the bonus 3 extension marks, we believe we deserve a solid 2 out of the 3 bonus marks due to going above and beyond with our focus on the test cases and realistic spreadsheet behaviour. Travis-Cl helped out tremendously through continuous integration however this took a long time to set up and integrate properly.
To go along with the Travis extension, we have also implemented a basic string builder which allows the formation of sentences within a cell.
Finally, we intend to finish implementing a substitution tail – something we ran out of time doing – after the assignment has been handed in.