Section1:

To keep the document concise, I will add the grammar files separately to the zip folder.
A version for each format will be submitted: EBNF , LL1 in a .grm .grm.atocc .grm.ucalgary

Section2:

First and Follow sets will be also added as seperate files in .follow and .first files

Section3:

To solve this problem I implemented a table-driven parser.
After transforming the grammar into LL(1) manually, I used the .grm file to extract the rules and store them in Python Dictionaries with two keys: “LHS” and “RHS” representing the left hand side and the right hand side of the rule respectively.
After extracting the rules, I extracted the first and follow sets from the respective files and stored them in Dictionaries as well.
A Grammar module is responsible for extracting all the data mentioned above.
In the Parser module, I implemented a method to generate the parsing table as well as a method to display it in a HTML format for debugging purposes. After that, I implemented the parse method as well as the skip error method as shown in the lecture slides.
During the parsing, important messages are written to the derivation file to show the derivation of the program.
The skipError() method was responsible for writing error information to the error.txt file .

Section4:

The tools used include the grammartool.jar provided by Professor Paquet, the atocc windows tool, the Ucalgary webapp