The project should be implemented in Java using Oracle's Java Development Kit (JDK, http://www.oracle.com/technetwork/java/javase/downloads/) or some other Java development environment. I hope you already download and installed on your systems.

Level 1: Scanning and Parsing

In this part of the project you will implement a scanner and a recursive descent parser for MicroJava. Start with the implementation of the scanner and do the following steps:

1. Study the specification of MicroJava carefully (Appendix A download from student portal). What are the tokens of the MicroJava grammar? What is the syntax of identifiers, numbers, character constants and comments? What keywords and predeclared names do you need? study careful
2. First step to create a package *MJ* and download the files *Scanner.java* and *Token.java* from into this MJ package. Look at those files and try to understand what they do.
3. Complete the skeleton file *Scanner.java* according to the slides of the course, check again if you have any confusion, and compile *Token.java* and *Scanner.java*.
4. Download the file *TestScanner.java* into the package MJ and compile it.
5. Download the MicroJava program *sample.mj* and run *TestScanner* on it.
6. Download the MicroJava program *BuggyScannerInput.mj* and run *TestScanner* on it in order to check if incorrect tokens are handled properly.

Case II: Next, you should write a recursive descent parser that uses your scanner to read and check the input tokens. Do the following steps given below:

1. Download the file *Parser.java* into the package *MJ* and check and observe what it does.
2. Complete the skeleton file *Parser.java* according to the slides of the course, check again if you have any confusion. Write a recursive descent parsing method for every production of the MicroJava grammar (see Appendix A). Compile *Parser.java*.
3. Download the file *TestParser.java*, compile it, and run it on *sample.mj*. If your parser is correct no errors should be reported.
4. Extend *Parser.java* with an error recovery according to the slides of the course. Add synchronisation points at the beginning of statements and declarations.
5. Download the MicroJava program *BuggyParserInput.mj* and run *TestParser* on it in order to check if syntax errors are handled correctly.