Group 4:

- **Dilan Shaminda**
- **Sadaf Alvani**
- **Server Khalilov**
- **Tareq Nasif**

Overall structure of the solution

- Recreated the grammar according to the provided rules in the assignment PDF file.
- In order to handle Methods and Variables declarations, we have created a separated class **MemberDecls,** which processes those cases.
- In grammar, in specific cases, we have some kind of recursion. It is like that for ***** cases, which includes **ClassDecl*, MemberDecl*, ParamRest*, BlockStatement*** and **ExprRest*.** In order to deal with them, we have followed the advice that was given in the exercise class and treated those ***** cases as single element together with a list. For example, for the **ParamRest*** we represent two cases:
  paramRestList ::= paramRest:pr paramRestList:rlist
  There are also optional cases (?), which do not require additional treatment.
- The operator precedence and associativity problems were solved by adding precedence rules, which look like the following piece of code:

  precedence right EQ;
  precedence left AND;
  precedence left EQUALS;
  precedence left LESS;
  precedence left PLUS, MINUS;
  precedence left TIMES, DIV;
  precedence right NEG, NEW;
  precedence left DOT, LBRACE, LBRACKET, LPAREN;

  The order of the precedence is from the bottom – highest to the top – lowest. Therefore, operations DOT, LBRACE, LBRACKET, LPAREN have the highest precedence. Afterwards, NEGATION and NEW have higher precedence than TIMES, DIV, which have the higher precedence than PLUS, MINUS accordingly. Moreover, LESS, EQUALS, AND and EQ are ordered as the following: LESS > EQUALS > AND > EQ.

  The keyword left state that we have defined the association from left to right and vice versa.

- In order to deal with incorrect expressions like the following:
  - 1+1; ()
  - x()=1; (left hand side: should be a variable (local, field access, or array access)

  we have created separate **Visitor** class, which extends **MJ.DefaultVisitor** and overrides visit methods with specific parameters. In the process, we also add the **new Syntax** errors to the list with a purpose to give appropriate information of the exceptions.
- Dealt with 2-dimensional arrays problem, handling it in the .cup file.
- Finally, we have added more test cases, which we think cover the code more.