A

Marker

Required language features as implemented and tested in the assignment		
Key Symbols	Cells	Notes
A	A	class declarations
A	A	data member declarations
A	A	member function declarations
A	A	inheritance list
A	A	private/public members
A	A	free function definitions
A	A	member function definitions
A	A	int, float, variable declarations
A	A	array variable declarations
A	A	if statement
A	A	while-statement
A	A	read/write-statement
A	A	return-statement
A	A	assignment statement
A	A	complex nested structures involving {} and (), and expressions as array indexes
A	A	complex expressions involving addop, multiop, relop, unarop, assnop, function calls

Notes	Tolerance	Cells	Values	
	100%	2.00%		**1 Document Section 1 - Presentation of LL(1) grammar, and realted transformations**
	100%	2.00%	A	1.1 Clarity - presentation of grammar and the problems/transformations
	100%	2.00%	A	1.2 Completeness - all problems with the grammar are listed/addressed
	100%	2.00%	A	1.3 Comment of changed specification versus original specification in the assignment statement (if applicable)
	100%	4.00%		**2 Document Section 2 - FIRST/FOLLOW sets**
	100%	2.00%	A	2.1 Clarity - presentation of FIRST/FOLLOW sets is clear and understandable
	100%	2.00%	A	2.2 Completeness - FIRST/FOLLOW sets are given for all non-terminal symbols
	100%	6.00%		**3 Document Section 3 - Description/rationale of the overall structure of the solution and the roles of the individual components used in the applied solution to the stated problem**
	100%	2.00%	A	3.1 Clarity - presentation of the design is clear and understandable
	100%	2.00%	A	3.2 Completeness - all components are identified and described in the design description
	100%	2.00%		**4 Document Section 4 - Description of tools/libraries/techniques used in the analysis/implementation. Description of other tools that might have been used. Justification of why the chosen tools were selected**
	100%	2.00%	A	4.1 All tools used in the analysis/implementation are mentioned
	100%	2.00%	A	4.2 Justification for all tools
	100%	20.00%		**5 Correct implementation of a top-down predictive parser according to the original grammar**
	100%	1.86%	A	5.1 class declarations
	100%	1.86%	A	5.2 data member declarations
	100%	1.86%	A	5.3 member function declarations
	100%	1.86%	A	5.4 inheritance list
	100%	1.86%	A	5.5 private/public members
	100%	1.86%	A	5.6 free function definitions
	100%	1.86%	A	5.7 member function definitions
	100%	1.86%	A	5.8 int, float, variable declarations
	100%	1.86%	A	5.9 array variable declarations
	100%	1.86%	A	5.10 if statement
	100%	1.86%	A	5.11 while-statement
	100%	1.86%	A	5.12 read/write-statement
	100%	1.86%	A	5.13 return-statement
	100%	1.86%	A	5.14 assignment statement
	100%	1.86%	A	5.15 complex nested structures involving {} and (), and expressions as array indexes
	100%	1.86%	A	5.16 complex expressions involving addop, multiop, relop, unarop, assnop, function calls
	100%	5.00%		**6 Output - outputs/errors file. Output of clear error messages (error description and location) in a file**
	100%	1.20%	A	6.1 Clarity - error messages describe what specific error was found
	100%	1.20%	A	6.2 Clarity - error messages describe what specific error was found
	100%	1.20%	A	6.3 Clarity - error messages include the erroneous lexeme that correspond to the error found
	100%	1.20%	A	6.4 Correctness - errors are correctly identified and correct syntax is not reported as errors
	100%	1.20%	A	6.5 Completeness - all errors are reported - even possible when error recovery is triggered
	100%	10.00%		**7 Output - outputs/errors file. Output of a derivation corresponding to the parsing**
	100%	5.00%	A	7.1 Clarity - derivation is somehow readable
	100%	5.00%	A	7.2 Correctness - output is truly a derivation in the strict sense of the term
	100%	5.00%		**8 Error recovery - the parser can recover from syntax errors**
	100%	5.00%	A	8.1 Error recovery is implemented - the parser can recover after an error
	100%	20.00%		**9 Test cases - completeness of testing (in addition to the provided files)**
	100%	1.50%	A	9.1 class declarations
	100%	1.50%	A	9.2 data member declarations
	100%	1.50%	A	9.3 member function declarations
	100%	1.50%	A	9.4 inheritance list
	100%	1.50%	A	9.5 private/public members
	100%	1.50%	A	9.6 free function definitions
	100%	1.50%	A	9.7 member function definitions
	100%	1.50%	A	9.8 int, float, variable declarations
	100%	1.50%	A	9.9 array variable declarations
	100%	1.50%	A	9.10 if statement
	100%	1.50%	A	9.11 while-statement
	100%	1.50%	A	9.12 read/write-statement
	100%	1.50%	A	9.13 return-statement
	100%	1.50%	A	9.14 assignment statement
	100%	1.50%	A	9.15 complex nested structures involving {} and (), and expressions as array indexes
	100%	1.50%	A	9.16 complex expressions involving addop, multiop, relop, unarop, assnop, function calls
	100%	5.00%		**10 Successful/correct use of tools/libraries/techniques in the analysis/implementation**
	100%	2.50%	A	10.1 The program never crashes or throws exceptions
	100%	2.50%	A	10.2 Tools presented in the labs are used appropriately, or comparable tools are used appropriately

Marker instructions:
Enter values only in the red cells. Everything else is calculated automatically.
In Column D, enter either A, B, C, or F for each marking element
Enter notes in column H. These notes will be then automatically uploaded on moodle as feedback to the students.

Designer instructions
Add/remove some lines with red cells to add/remove evaluation criteria
When you add lines, add them only in the middle of the area, or else the calculation is going to be wrong
To add lines, add a new row then copy into it one of the existing rows

Grading criteria	Cells		Score cells	Total marks	Indicators
Document Section 1 - Presentation of ll(1) grammar, and realted transformations	6	0.00	0.00	0.00	2.2
Document Section 2 - FIRST/FOLLOW sets	6	2.00	0.00	2.00	2.2
Document Section 3 - Description/rationale of the overall structure of the solution and the roles of the	6	2.00	0.00	2.00	4.3
Document Section 4 - Description of tools/libraries/techniques used in the analysis/implementation	6	2.00	0.00	2.00	5.2
Correct implementation of a top-down predictive parser according to the original grammar	96	19.00	0.00	19.00	4.4
Output - outputs/errors file. Output of clear error messages (error description and location) in a	6	3.00	0.00	3.00	4.4
Output - outputs/errors file. Output of a derivation corresponding to the parsing	6	3.00	0.00	3.00	4.4
Error recovery - the parser can recover from syntax errors	6	3.00	0.00	3.00	4.4
Test cases - completeness of testing (in addition to the provided files)	6	12.00	0.00	12.00	4.4
Successful/correct use of tools/libraries/techniques in the analysis/implementation	6	2.00	0.00	2.00	5.1
Total	6	0.00	0.00	0.00	

All notes
1 :
1.1 : A :
1.2 : A :
1.3 : A :
2 : :
2.1 : A :
2.2 : A :
3 : :
3.1 : A :
3.2 : A :
4 : :
4.1 : A :
4.2 : A :
5 : :
5.1 : A :
5.2 : A :
5.3 : A :
5.4 : A :
5.5 : A :
5.6 : A :
5.7 : A :
5.8 : A :
5.9 : A :
5.10 : A :
5.11 : A :
5.12 : A :
5.13 : A :
5.14 : A :
5.15 : A :
5.16 : A :
6 : :
6.1 : A :
6.2 : A :
6.3 : A :
6.4 : A :
6.5 : A :
7 : :
7.1 : A :
7.2 : A :
8 : :
8.1 : A :
9 : :
9.1 : A :
9.2 : A :
9.3 : A :
9.4 : A :
9.5 : A :
9.6 : A :
9.7 : A :
9.8 : A :
9.9 : A :
9.10 : A :
9.11 : A :
9.12 : A :
9.13 : A :
9.14 : A :
9.15 : A :
9.16 : A :
10 : :
10.1 : A :
10.2 : A :