Marker

Programming language features as implemented and tested in the assignment

Pass	Fail	N/A	
A			class declarations
A			data member declarations
A			member function declarations
A			inheritance list
A			private/public members
A			free function definitions
A			member function definitions
A			int, float, variable declarations
A			array variable declarations
A			if statement
A			while statement
A			read/write statement
A			return statement
A			assignment statement
A			complex nested structures involving () and [], and expressions as array indexes
A			complex expressions involving addop, multop, relop, unaryop, literal, function calls

Notes	Tokens	Lexlis	Index			
	100%	0.00%			**1 Document Section 1 - Presentation of LL(1) grammar, and realted transformations**	1 :
	100%	1.00%	A	1.1	Clarity - presentation of grammar and the problems/transformations	1.1 : A :
	100%	1.00%	A	1.2	Completeness - all problems with the grammar are listed/explained	1.2 : A :
	100%	1.00%	A	1.3	Statement of changed specification versus original specifications in the assignment statement (if applicable)	1.3 : A :
	100%	0.00%			**2 Document Section 2 - FIRST/FOLLOW sets**	2 :
	100%	2.00%	A	2.1	Clarity - presentation of FIRST/FOLLOW sets is clear and understandable	2.1 : A :
	100%	2.00%	A	2.2	Completeness - FIRST/FOLLOW sets are given for all non-terminal symbols	2.2 : A :
	100%	5.00%			**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**	3 :
	100%	2.00%	A	3.1	Clarity - presentation of the design is clear and understandable	3.1 : A :
	100%	2.00%	A	3.2	Completeness - all components are described in the design description	3.2 : A :
	100%	4.00%			**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**	4 :
	100%	2.00%	A	4.1	All tools used in the analysis/implementation are mentioned	4.1 : A :
	100%	2.00%	A	4.2	Justification for all tools	4.2 : A :
	100%	20.00%			**5 Correct implementation of a top-down predictive parser according to the original grammar**	5 :
	100%	1.80%	A	5.1	Class declarations	5.1 : A :
	100%	1.80%	A	5.2	Data member declarations	5.2 : A :
	100%	1.80%	A	5.3	Member function declarations	5.3 : A :
	100%	1.80%	A	5.4	Inheritance list	5.4 : A :
	100%	1.80%	A	5.5	Private/public members	5.5 : A :
	100%	1.80%	A	5.6	Free function definitions	5.6 : A :
	100%	1.80%	A	5.7	Member function definitions	5.7 : A :
	100%	1.80%	A	5.8	Int, float, variable declarations	5.8 : A :
	100%	1.80%	A	5.9	Array variable declarations	5.9 : A :
	100%	1.80%	A	5.10	If statement	5.10 : A :
	100%	1.80%	A	5.11	While statement	5.11 : A :
	100%	1.80%	A	5.12	Read/write statement	5.12 : A :
	100%	1.80%	A	5.13	Return statement	5.13 : A :
	100%	1.80%	A	5.14	Assignment statement	5.14 : A :
	100%	1.80%	A	5.15	Complex nested structures involving () and [], and expressions as array indexes	5.15 : A :
	100%	1.80%	A	5.16	Complex expressions involving addop, multop, relop, unaryop, literal, function calls	5.16 : A :
	100%	10.00%			**6 Output - outputtoken.txt file: Output of clear error messages (error description and location) in a file**	6 :
	100%	1.20%	A	6.1	Completeness - all errors are reported - even after error recovery is triggered	6.1 : A :
	100%	1.20%	A	6.2	Clarity - error messages describe what specific kind of error was found	6.2 : A :
	100%	1.20%	A	6.3	Clarity - error messages include the erroneous lexeme that correspond to the error found	6.3 : A :
	100%	1.20%	A	6.4	Correctness - errors are correctly identified and correct syntax is not reported as errors	6.4 : A :
	100%	1.20%	A	6.5	Completeness - all errors are reported - even after error recovery is triggered	6.5 : A :
	100%	10.00%			**7 Output - outputderivation.txt file: Output of a derivation corresponding to the parsing**	7 :
	100%	5.00%	A	7.1	Clarity - derivation is somehow readable	7.1 : A :
	100%	5.00%	A	7.2	Correctness - output is really a derivation in the strict sense of the term	7.2 : A :
	100%	5.00%			**8 Error recovery – the parser can recover from syntax errors**	8 :
	100%	5.00%	A	8.1	Error recovery is implemented - the parser can recover after an error	8.1 : A :
	100%	20.00%			**9 Test cases - completeness of testing (in addition to the provided files)**	9 :
	100%	1.50%	A	9.1	Class declarations	9.1 : A :
	100%	1.50%	A	9.2	Data member declarations	9.2 : A :
	100%	1.50%	A	9.3	Member function declarations	9.3 : A :
	100%	1.50%	A	9.4	Inheritance list	9.4 : A :
	100%	1.50%	A	9.5	Private/public members	9.5 : A :
	100%	1.50%	A	9.6	Free function definitions	9.6 : A :
	100%	1.50%	A	9.7	Member function definitions	9.7 : A :
	100%	1.50%	A	9.8	Int, float, variable declarations	9.8 : A :
	100%	1.50%	A	9.9	Array variable declarations	9.9 : A :
	100%	1.50%	A	9.10	If statement	9.10 : A :
	100%	1.50%	A	9.11	While statement	9.11 : A :
	100%	1.50%	A	9.12	Read/write statement	9.12 : A :
	100%	1.50%	A	9.13	Return statement	9.13 : A :
	100%	1.50%	A	9.14	Assignment statement	9.14 : A :
	100%	1.50%	A	9.15	Complex nested structures involving () and [], and expressions as array indexes	9.15 : A :
	100%	1.50%	A	9.16	Complex expressions involving addop, multop, relop, unaryop, literal, function calls	9.16 : A :
	100%	0.00%			**10 Successful/correct use of tools/libraries/techniques in the analysis/implementation.**	10 :
	100%	1.00%	A	10.1	The program never crashes or throws exceptions	10.1 : A :
	100%	1.00%	A	10.2	Tools presented in the labs are used appropriately, or comparable tools are used appropriately	10.2 : A :

Marker instructions

Enter values only in the red cells. Everything else is calculated automatically.
In Column D, enter either A, R, C, or F for each marking element
Enter notes in column A. 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 cluster	ratio		Score ratio	Total marks	Indicators
Document Section 1 - Presentation of ll(1) grammar, and realted transformations	5	0.00	5.00	0.00	2.3
Document Section 2 - FIRST/FOLLOW sets	5	2.00	5.00	4.00	2.3
Document Section 3 - Description/rationale of the overall structure of the solution and the roles of	5	2.00	5.00	2.00	4.3
Document Section 4 - Description of tools/libraries/techniques used in the analysis/implementation	5	2.00	5.00	4.00	5.3
Correct implementation of a top-down predictive parser according to the original grammar	75	15.00	5.00	15.00	4.4
Output - outputtoken.txt file: Output of clear error messages (error description and location) in a	5	5.00	5.00	5.00	4.4
Output - outputderivation.txt file: Output of a derivation corresponding to the parsing	5	5.00	5.00	5.00	4.4
Error recovery – the parser can recover from syntax errors	5	5.00	5.00	5.00	4.4
Test cases - completeness of testing (in addition to the provided files)	5	12.00	5.00	12.00	4.4
Successful/correct use of tools/libraries/techniques in the analysis/implementation.	5	5.00	5.00	5.00	5.3
Total	95		95	52.00	

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 :