

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI
DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION SYSTEMS

Compiler Construction (CS F363)
II Semester 2021-22
Compiler Project (Stage-1 Submission)
Coding Details
(March 3, 2022)

Group Number

10

1. Team Members Names and IDs

ID	2019A7PS0022P	Name	<u>Harsh Butani</u>
ID	2019A7PS0033P	Name	<u>Siddharth Upadhyay</u>
ID	2019A7PS0070P	Name	<u>Hemant Singh Sisodiya</u>
ID	2019A7PS0100P	Name	<u>Mohit Sharma</u>
ID	2019A7PS0294P	Name	<u>Akash S Revankar</u>

2. Mention the names of the Submitted files :

1. grammar.txt	6. parser.h	11. set.c	16. parser.c	21. customtest.txt
2. lexerDef.h	7. set.h	12. stack.c	17. hashtable.c	22. testcase1.txt
3. parserDef.h	8. hashtable.h	13. driver.c	18. makefile	23. testcase2.txt
5. driver.h	9. stack.h	14. tree.c	19. tokens.txt	24. testcase3.txt
6. lexer.h	10. tree.h	15. lexer.c	20. nonTerminals.txt	25. testcase4.txt
				26. testcase5.txt

3. Total number of submitted files (including copy the pdf file of this coding details pro forma) : **27** (All files should be in ONE folder named as Group_#)

4. Have you compressed the folder as specified in the submission guidelines? (yes/no) **Yes**

5. **Lexer Details:**

- [A]. Technique used for pattern matching: **DFA implemented using switch case**
- [B]. Keyword Handling Technique: **Inserted the keywords in a lookup table (implemented using hashtable)**
- [C]. Hash function description, if used for keyword handling: **Polynomial Rolling Hash Function**
- [D]. Have you used twin buffer? (yes/ no): **Yes**
- [E]. Error handling and reporting (yes/No): **Yes**
- [F]. Describe the errors handled by you: **Length exceeded, Invalid Token**
- [G]. Data Structure Description for tokenInfo (in maximum two lines):
We have used a structure consisting of following members – int for line number. , enum for tokens, and union for value of TK_ID,TK_NUM,TK_RNUM

6. **Parser Details:**

[A]. High Level Data Structure Description (in maximum three lines each, avoid giving C definitions used):

- i. grammar : **Array of linkedlist**
- ii. FIRST and FOLLOW sets: **Array of Int**
- iii. parse table: **2D array**
- iv. parse tree: (Describe the node structure also): **Parse tree is an n-ary tree.**
 - **Node structure: pointers to parent, leftmost child, rightmost child and brother, number of children, symbol, token.**
 - **Symbol – union for enum of terminal and nonterminal, bool for isTerminal**
- v. Any other (specify and describe)

We will execute the code after running make as follows - ./stage1exe <input filename> <output filename>

[B]. Parse tree

- i. Constructed (yes/no): **Yes**
- ii. Printing as per the given format (yes/no): **Yes**
- iii. Describe the order you have adopted for printing the parse tree nodes (in maximum two lines):
We are printing in the format - inorder in file, and preorder on console

[C]. Grammar and Computation of First and Follow Sets

- i. Data structure for original grammar rules: **LinkedList**
- ii. FIRST and FOLLOW sets computation automated (yes /no): **Yes**
- iii. Name the functions (if automated) for computation of First and Follow sets: **computeFirstSets, computeFollowSets**

[D]. If computed First and Follow sets manually and represented in file/function (name that) **N.A.**

[E]. Error Handling

- iv. Attempted (yes/ no): **Yes**
- v. Describe the types of errors handled: **Unknown Symbol, Invalid Token, Stack Top not matching with Input Symbol**

7. Compilation Details:

- [A]. Makefile works (yes/no): **Yes**
[B]. Code Compiles (yes/ no): **Yes**
[C]. Mention the .c files that do not compile: **None**
[D]. Any specific function that does not compile: **None**
[E]. Ensured the compatibility of your code with the specified gcc version (yes/no) **Yes**

8. Driver Details: Does it take care of the options specified earlier (yes/no): **Yes**

9. Execution

- [A]. status (describe in maximum 2 lines): **Every file compiles. lexer, parser gives correct output for all test cases (Except parser gives partially correct output for t6.txt).**
[B]. Gives segmentation fault with any of the test cases (1-6) uploaded on the course page. If yes, specify the testcase file name: **None**

10. Specify the language features your lexer or parser is not able to handle (in maximum one line): **Lexer handles everything, whereas parser partially handles error recovery. Else parser is correct.**

11. Are you availing the lifeline (Yes/No): **No**

12. Declaration: We, **Harsh Butani, Sidhharth Upadhyay, Hemant Singh Sisodiya, Mohit Sharma, Akash S Revankar** declare that we have put our genuine efforts in creating the compiler project code and have submitted the code developed only by us. We have not copied any piece of code from any source. If our code is found plagiarized in any form or degree, we understand that a disciplinary action as per the institute rules will be taken against all of us in our team and we will accept the penalty as decided by the department of Computer Science and Information Systems, BITS, Pilani.

Your names and IDs

ID 2019A7PS0022P
ID 2019A7PS0033P
ID 2019A7PS0070P
ID 2019A7PS0100P
ID 2019A7PS0294P

Name Harsh Butani
Name Siddharth Upadhyay
Name Hemant Singh Sisodiya
Name Mohit Sharma
Name Akash S Revankar

Date: **03-03-2022**
