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

Compiler Construction (CS F363)
II Semester 2024-25
Compiler Project
Coding Details
(March 15, 2025)

Group Number
14

1. Team Members Names and IDs
   ID _2022A7PS0013P_____Name_Vikram Hariharan_____
   ID _2021B5A71159P_____Name_Ankur Renduchintala_____
   ID _2021B4A71134P_____Name_Abhiram H_____
   ID _2021B4A71111P_____Name_Avyakth Krishnakumar_____
   ID _2021B1A72281P_____Name_Suchit Chebolu_____
   ID _____Name_____
2. Mention the names of the Submitted files :

1 coding details.pdf	7 lexerDef.h	13 tester.c
2 driver.c	8 makefile	14 test lexer.c
3 grammar.txt	9 lexer.c	15 stack.h
4 hash map.c	10 parser.h	16 stack.c
5 hash map.h	11 parserDef.h	17 symbolTable.c
6 lexer.h	12 parser.c	18 symbolTable.h

3. Total number of submitted files (including copy the pdf file of this coding details pro forma): 31
   (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: Finite state machine (FSM) or deterministic finite automaton (DFA) when the transitions are uniquely determined.
[B]. Keyword Handling Technique: Using hash functions
[C]. Hash function description, if used for keyword handling: djb2 and a variant of djb2
[D]. Have you used twin buffer? (yes/ no): Yes
[E]. Error handling and reporting (yes/No): Yes
[F]. Describe the errors handled by you: Unknown Symbol, unknown pattern, length check for function id and identifier
[G]. Data Structure Description for tokenInfo (in maximum two lines): _Structure named token with parameters: enum TokenType name, char* lexeme, int line_num, Union of long long num and long double r_num , bool is_value_int._____
_____
_____

6. Parser Details:

[A]. High Level Data Structure Description (in maximum three lines each, avoid giving C definitions used):
i. grammar : ___structure NonTerminal, with parameters: char* name, int prod_count, int first_size, int follow_size, production** productions, token_type* first_set, token_type* follow_set, bool has_epsilon_and_first,
_____
_____

ii. FIRST and FOLLOW sets _stored within struct NonTerminal as an array

iii. parse table __HashMap with struct Entry* entries, int capacity, int size_, struct Entry has unsigned int key, struct production* value, bool is_occupied

iv. parse tree: (Describe the node structure also) Node is a structure with parameters symbol* stack_symbol , token* token_value , node** children, int children_count

v. Any other (specify and describe) _____

[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) __in order traversal: prints left most child followed by parent followed by right siblings

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

i. Data structure for original grammar rules _____
ii. FIRST and FOLLOW sets computation automated (yes /no)__yes_____
iii. Name the functions (if automated) for computation of First and Follow sets_add_to_first_set , add_to_follow_set , compute_first_set , compute_follow_set , compute_first_of_sequence , contains_EPS , add_to_set _____
iv. If computed First and Follow sets manually and represented in file/function (name that) _____

[D]. Error Handling

v. Attempted (yes/ no):_Yes_____
vi. Describe the types of errors handled __unknown symbol, unknown pattern, variable length mismatch, function length mismatch_____

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):___Successful execution without segmentation faults. Memory is freed at the end of execution._____

[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)__none_____
    _____

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

12. Declaration: We, _Ankur Renduchintala, Abhiram H, Avyakth K, Suchit Chebolu, Vikram Hariharan_____ (your names) 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
    Name:_ Ankur Renduchintala _____ID: __2021B5A71159P _____
    Name:_ Abhiram H,_____ID: __2021B4A71134P _____
    Name:__ Avyakth K _____ID: ___2021B4A71111P _____
    Name:__ Suchit Chebolu _____ID: __2021B1A72281P _____
    Name:__ Vikram Hariharan _____ID: __2022A7PS0013P _____

    Name:_____ID: _____
    Date: _15 March 2025_____

    Not to exceed 3 pages.