**SOFTWARE CONSTRUCTION (PRACTICALS) – SPRING 2013**

**EXPERIMENT 3 – LEXICAL ANALYSIS**

|  |  |  |  |
| --- | --- | --- | --- |
| **DATE: \_\_\_\_\_\_\_\_\_\_\_\_\_\_** | | **Student Names: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** | |
| **Marks Obtained: \_\_\_\_\_\_\_\_** | | **COURSE: BESE 16 \_\_\_\_\_** | |
| **Deadline: 1400 hrs 18 Feb 2013** | | **Instructor: Engr. Umar Mahmud** | |
|  | **Instructions**   * This is a syndicate effort. At most **THREE** per syndicate. * Plagiarism is strictly forbidden. * Write your remarks next to the space provided. Submit hard copy of the report before deadline. Marks will be deducted for late submissions. | |  |
| 1. | **Objectives:**   1. Learning the Lexical Analysis 2. Tokenizing 3. Identifying tokens into keywords and operators 4. Creating a literal table | |  |
| 2. | **Time Required:** 3 hrs | |  |
| 3. | **Programming Language:** Java/C++/C# | |  |
| 4. | **Software Required:**   1. Windows OS 2. C++/Java Compiler | |  |
| 5. | **Lexical Analysis:** A program written in a language is a file of strings. The lexical analysis converts the strings into tokens. The program that does lexical analysis is called as a lexer or scanner. | |  |
| 6. | **Tokens:** Tokens are strings that have a meaning in some programming context. For the expression **int i = 3 +2;** the tokenized output is as follows   1. int keyword 2. i variable/identifier 3. = assignment operator 4. 3 Integer literal 5. + summation operator 6. 2 Integer literal 7. ; end of statement | |  |
| 7. | For every statement a corresponding list/table of literals is created. | |  |
| 8. | **Task 1 – Basic Scanning:** For the first task create the statement **‘int j = 23 / 40 ;’** as a string and tokenize it. Every literal is separated with a space character. Using StringTokenizer class of Java will be easier. Output the literal table (i.e., show all tokens) and print the outcome here (in table format similar to as shown in Point 6): - | | (2) |
| 9. | **Task 2 - Identifying Keywords:** Create definitions of keywords through nested IF-ELSE statements for this task. Take input from user and identify if the given input is a keyword. The standard keywords that you may use are as follows :-  class void main int float char if else for return | |  |
| 10. | **Recognizing Numbers and Words:** Since the language of numbers and words is infinite and we need to know whether a given item is a valid number or a word. Create a code that identifies whether a given item is a number or string e.g., abcdef is a string, 32 is an integer and 3.14 is a float. You may take input from the user. | |  |
| 11. | What will be the output if the user types in 32try.9i? Did your program give the correct output? Why/Why not? | | (2) |
| 12. | **Special Characters and Operators:** To recognize special characters and operators use the following specification: -  **Operators:**  **+ - \* / =**  **== != > >= <**  **<= ++ -- += -=**  **Special Characters:**  **( ) [ ] #**  **{ } ; ,**  You may take input from the user. | |  |
| 13. | Now include Points 9, 10 and 12 in your code and create the new literal table for the statement in Point 6. | | (1) |
| 14. | Re-execute for the following statements and display the literal tables   1. double myDouble = 32.1009 ; 2. int myVariable = 67 ; | | (2) |
| 15. | Re-execute for the following statements and display the literal tables   1. int = j 32 ; 2. char c = a ; | | (2) |
| 16. | From your knowledge of programming do you find any errors in the statements of Point 15? If so, how would you remove it? | | (1) |