RA1911033010114

UNNATI MISHRA

LAB EXPERIMENT 1:

**IMPLEMENTATION OF LEXICAL ANALYZER**

**AIM:** To write a program to implement Lexical Analyzer.

**ALGORITHM:**

1. Start
2. Get the input program from the file program.txt
3. Read the program line by line and check if each word in a line is a keyword, identifier, math operator, logical operator, numerical value, or another symbol.
4. For each lexeme read, generate a token as follows:
5. If the lexeme is a keyword, then the token is the keyword itself
6. If the lexeme is an identifier, then the token generated is printed on the console as an identifier.
7. In the same way, the math operator, logical operator, numerical values and other symbols are printed on the console.
8. The stream of tokens generated is displayed in the console output.
9. Stop

**PROGRAM:**

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<ctype.h>

#include<vector>

int isKeyword(char buffer[]){

char keywords[40][20] = {"class","string","main","auto","using","namespace","include","break","case","char","const","continue","default",

"do","double","else","enum","extern","float","for","goto",

"if","int","long","public","register","return","short","signed",

"sizeof","static","struct","switch","typedef","union",

"unsigned","void","volatile","while"};

int i, flag = 0;

for(i = 0; i < 32; ++i){

if(strcmp(keywords[i], buffer) == 0){

flag = 1;

break;

}

}

return flag;

}

char isHeader(char buffer[]){

char headers[2][10] = {

"iostream", "stdio"

};

int i, flag = 0;

for(i = 0; i < 2; ++i){

if(strcmp(headers[i], buffer) == 0){

flag = 1;

break;

}

}

return flag;

}

int main(){

char ch, buffer[15], operators[] = "+-\*/%=<>", special[] = "#;,.{}[]()";

FILE \*fp;

int i,j=0;

fp = fopen("sample.txt","r");

if(fp == NULL){

printf("error while opening the file\n");

exit(0);

}

while((ch = fgetc(fp)) != EOF){

for(i = 0; i < 8; ++i){

if(ch == operators[i])

printf("%c is operator\n", ch);

}

for(i = 0; i < 10; ++i){

if(ch == special[i])

printf("%c is special character\n", ch);

}

if(isdigit(ch)==true)

printf("%c is an integer\n",ch);

if(isalnum(ch)){

buffer[j++] = ch;

}

else if((ch == ' ' || ch == '\n') && (j != 0)){

buffer[j] = '\0';

j = 0;

if(isKeyword(buffer) == 1)

printf("%s is keyword\n", buffer);

else if(isHeader(buffer)==1)

printf("%s is Header file\n", buffer);

else

printf("%s is identifier\n", buffer);

}

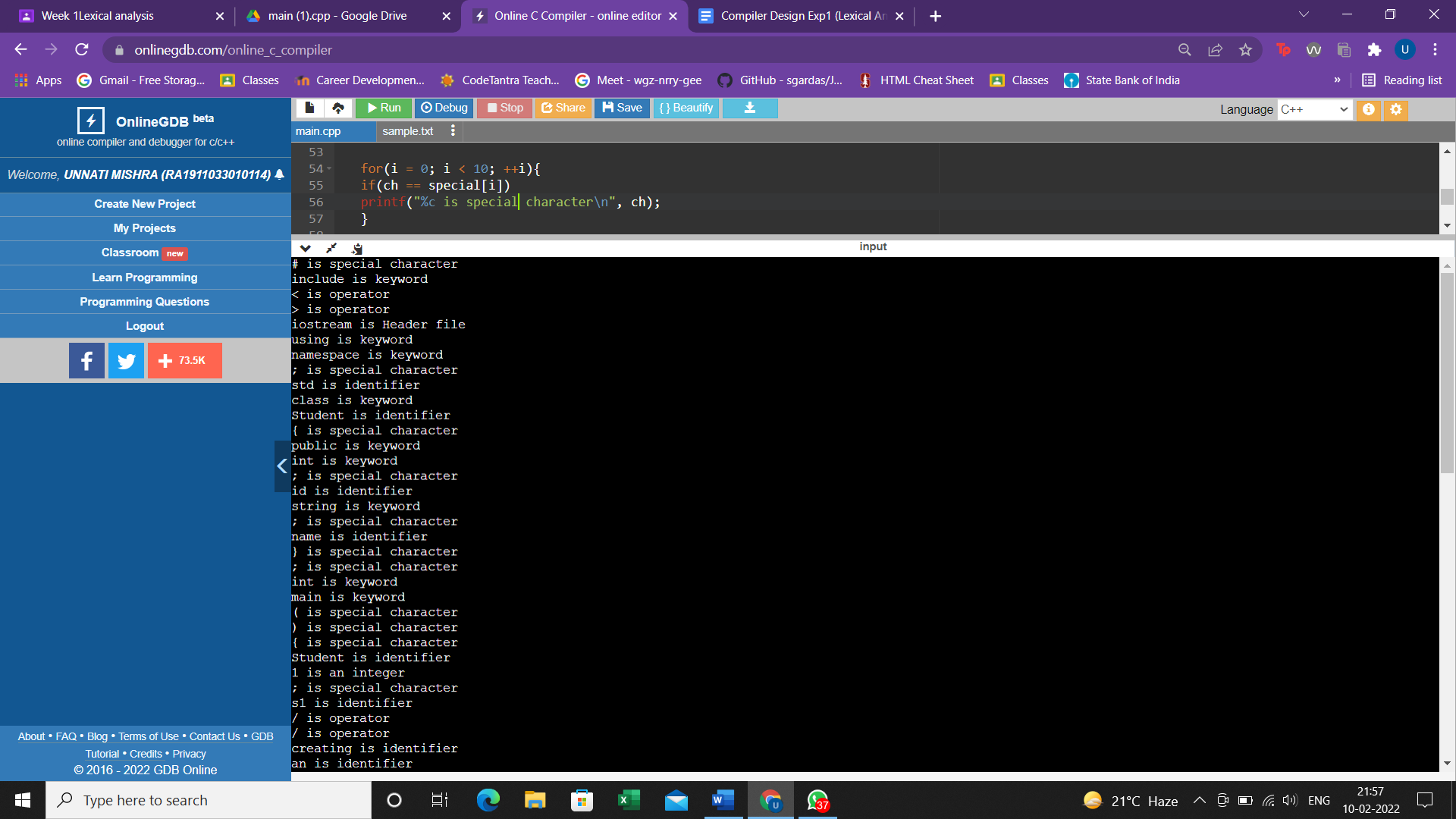
}

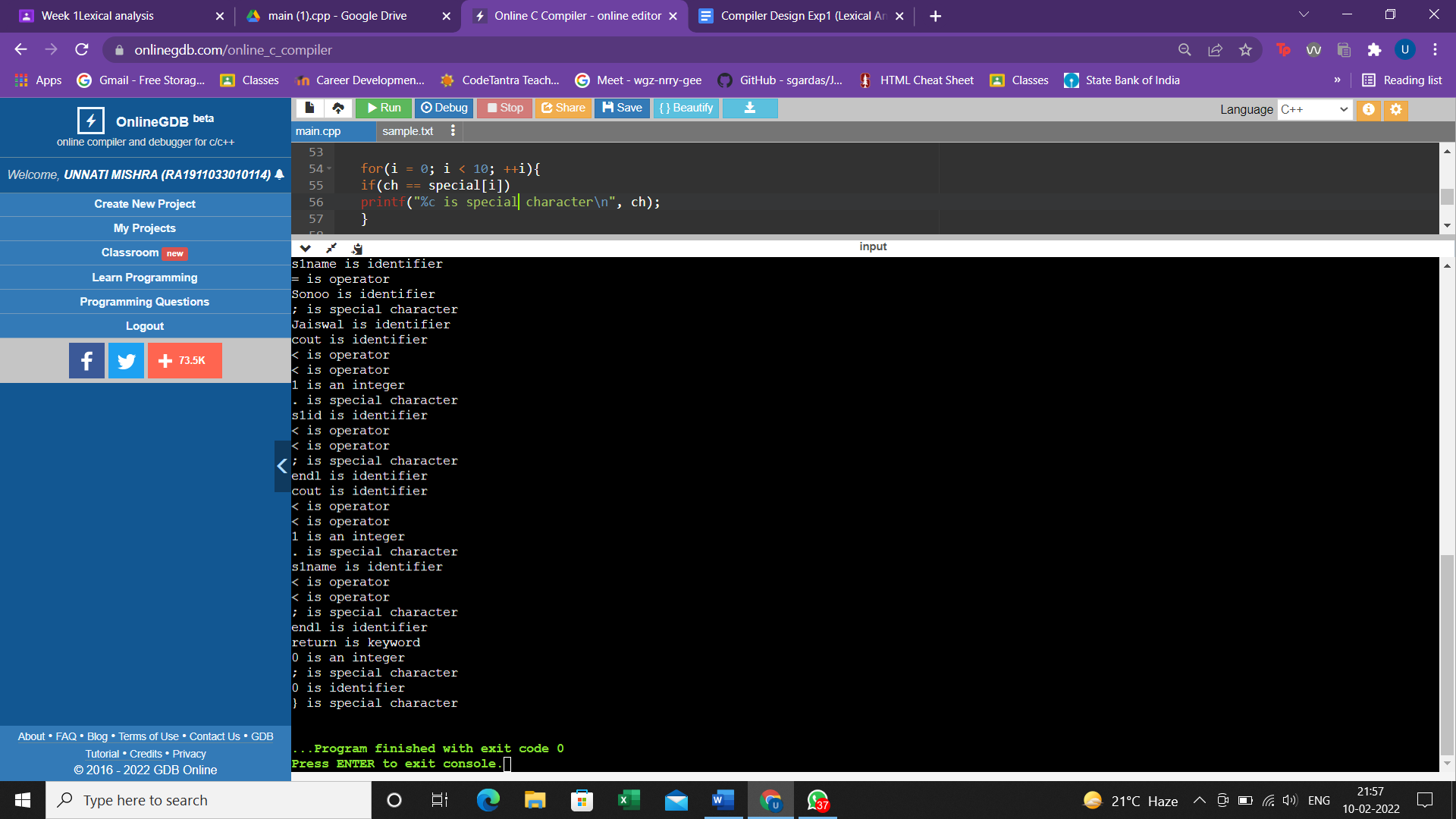
fclose(fp);

return 0;

}

**OUTPUT:**





**RESULT:** The implementation of the lexical analyzer in C++ was compiled, executed and verified successfully**.**