**Name: Mohammad Awais**

**Class: BSCS-8-A**

**CMS: 242554**

# Compiler Construction Lab 9

## Task 1

### < Code >

#include <iostream>

using namespace std;

#include <vector>

#include<string>

#include <typeinfo>

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* DECLARATIONS \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

string input;

vector<char> num\_list = {'0','1','2','3','4','5','6','7','8','9','.'};

int main(){

cout<<"\n\t \*\*\*\*\*\*\*\*\*\* TASK 1 \*\*\*\*\*\*\*\*\*\*\*\*\* \n\n";

cout<<"[?] Enter any value : ";

getline(cin, input);

int mode = -1; *// 0 : STRING, 1: INT, 2: FLOAT, 3: DOUBLE*

for(char c : input){

int num\_check = 0;

for(char n : num\_list){

if (c == n){

num\_check = 1;

if(n == '.' && mode !=2){

mode =2;

}

else if(n == '.' && mode == 2){

num\_check = 0 ;

}

break;

}

}

if(num\_check == 0){

mode=0;

break;

}

}

if(mode == 0){

string value = input;

cout<<"\n\t[+] Value Entered : "<<value<<endl;

cout<<"\n\t[+] Value Type : String ("<<typeid(value).name()<<")"<<endl;

}

else if (mode != 2){

int value = stoi(input);

cout<<"\n\t[+] Value Entered : "<<value<<endl;

cout<<"\n\t[+] Value Type : Int ("<<typeid(value).name()<<")"<<endl;

}

else{

string partA = "";

string partB = "";

int bCounter = 0;

int gear = 0;

for(char c : input){

if(gear == 0){

if(c == '.'){

gear =1;

continue;

}

partA = partA + c;

}

else if ( gear == 1){

partB = partB + c;

bCounter++;

}

}

if (partA.length()==0){

partA+='0';

}

if (partB.length()==0){

partB+='0';

}

if(bCounter > 7){

double value = stod(partA+"."+partB);

cout<<"\n\t[+] Value Entered : "<<value<<endl;

cout<<"\n\t[+] Value Type : Double ("<<typeid(value).name()<<")"<<endl;

}

else{

float value = stof(partA+"."+partB);

cout<<"\n\t[+] Value Entered : "<<value<<endl;

cout<<"\n\t[+] Value Type : Float ("<<typeid(value).name()<<")"<<endl;

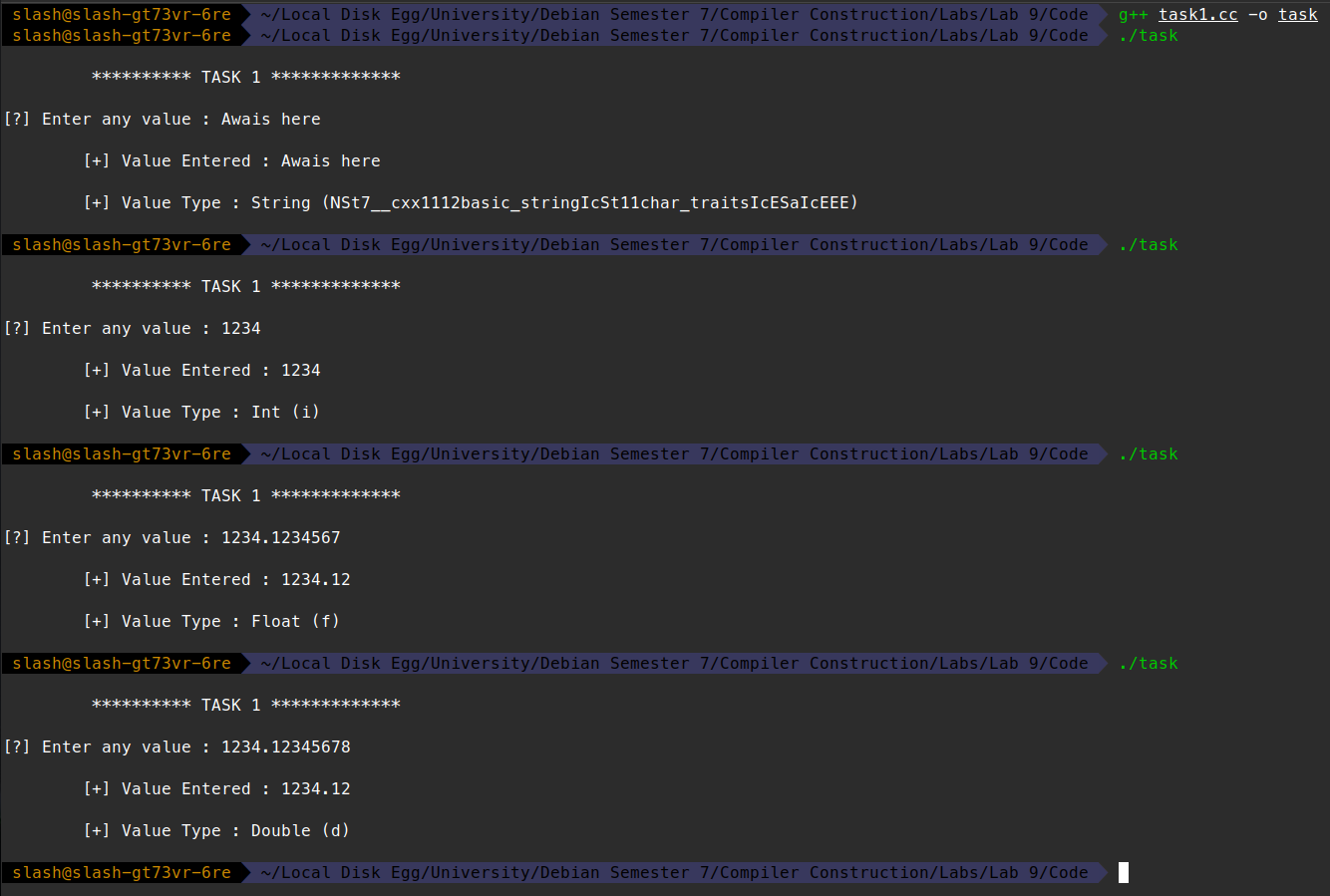
}

}

cout<<endl;

}

### < Output >



## Task 2

### < Code >

#### Scope.y

%{

#include <string>

#include <iostream>

#include <vector>

#include <array>

#include <math.h>

#define YYDEBUG 1

void Push();

std::vector<std::vector<int>> Pop();

std::vector<std::vector<int>> Top();

std::vector<std::string> id\_list = {};

void newIdentifier(std::string id\_name, int id\_value);

void Identifier(std::string id\_name, int id\_value);

int returnIdentifier(std::string id\_name);

int yylex();

void yyerror();

extern FILE \* yyin;

%}

%union {

int val;

char \*name;

};

%token keyword\_int

%token T\_int

%token identifier

%token cout

%%

S : E { };

E : keyword\_int identifier '=' T\_int ';' {newIdentifier(yylval.name, yylval.val);}

| keyword\_int identifier';' {newIdentifier(yylval.name, NULL);}

| identifier '=' T\_int ';' {Identifier(yylval.name, yylval.val);}

| cout '<''<' identifier ';'{std::cout<<">> "<<yylval.name<<" = "<<returnIdentifier(yylval.name)<<std::endl;}

| '{' E '}' { Push();}

| '}' E {Pop();}

%%

const int MAXSIZE = 40;

static std::array<std::vector<std::vector<int>>,MAXSIZE> stack;

void Push(){

std::vector<std::vector<int>> data = {};

for (int i = MAXSIZE-1; i>0;i--){

stack[i]=stack[i-1];

}

stack[0] = data;

}

std::vector<std::vector<int>> Pop(){

std::vector<std::vector<int>> data = stack[0];

for (int i = 0; i<MAXSIZE;i++){

stack[i]=stack[i+1];

}

return data;

}

std::vector<std::vector<int>> Top(){

return stack[0];

}

void newIdentifier(std::string id\_name, int id\_value){

int detect = 0;

int target\_id = -1;

for( int i = 0; i<id\_list.size() ; i++){

if(id\_name==id\_list[i]){

detect = 1;

target\_id = i;

break;

}

}

if(detect == 0){

id\_list.push\_back(id\_name);

target\_id = id\_list.size();

}

detect = 0;

for ( int i = 0; i<stack[0].size(); i++){

if(stack[0][i][0]==target\_id){

detect = 1;

break;

}

}

if(detect==1){

// ERROR BECAUSE TWO DECLARATIONS IN SAME SCOPE

std::cout<< " GIVE ERROR YOU FOOL [NEW]";

}

else{

stack[0].push\_back({target\_id,id\_value});

}

}

void Identifier(std::string id\_name, int id\_value){

int detect = 0;

int target\_id = -1;

for( int i = 0; i<id\_list.size() ; i++){

if(id\_name==id\_list[i]){

detect = 1;

target\_id = i;

break;

}

}

if(detect == 0){

// ERROR BECAUSE TWO DECLARATIONS IN SAME SCOPE

std::cout<< " GIVE ERROR YOU FOOL [ SIMPLE ]";

}

detect = 0;

int detect\_j = -1;

int detect\_i = -1;

for(int j = 0; j<stack.size();j++){

for ( int i = 0; i<stack[j].size(); i++){

if(stack[j][i][0]==target\_id){

detect = 1;

detect\_j = j;

detect\_i = i;

break;

}

}

if(detect == 1){

break;

}

}

if(detect==1){

stack[detect\_j][detect\_i][1]=id\_value;

}

else{

// ERROR BECAUSE TWO DECLARATIONS IN SAME SCOPE

std::cout<< " GIVE ERROR YOU FOOL [SIMPLE 2]";

}

}

int returnIdentifier(std::string id\_name){

int detect = 0;

int target\_id = -1;

for( int i = 0; i<id\_list.size() ; i++){

if(id\_name==id\_list[i]){

detect = 1;

target\_id = i;

break;

}

}

if(detect == 0){

// ERROR BECAUSE TWO DECLARATIONS IN SAME SCOPE

std::cout<< " GIVE ERROR YOU FOOL [ RETURN SIMPLE ]";

}

detect = 0;

int detect\_j = -1;

int detect\_i = -1;

for(int j = 0; j<stack.size();j++){

for ( int i = 0; i<stack[j].size(); i++){

if(stack[j][i][0]==target\_id){

detect = 1;

detect\_j = j;

detect\_i = i;

break;

}

}

if(detect == 1){

break;

}

}

if(detect!=1){

// ERROR BECAUSE TWO DECLARATIONS IN SAME SCOPE

std::cout<< " GIVE ERROR YOU FOOL [RETRUN SIMPLE 2]";

}

return stack[detect\_j][detect\_i][1];

}

int main() {

//return yyparse();

FILE \*file = fopen("testcode","r");

if(!file){

printf("[-] Can't open the file!\n");

return -1;

}

yyin=file;

//while(yyparse());

do

{

yyparse();

} while (!feof(yyin));

fclose(file);

printf("\n");

return 0;

}

#### Scope.l

%{

#include "scope.tab.h"

%}

%%

[0-9]+ { yylval.val = atoi(yytext); return T\_int;}

[=;{}<] { return yytext[0];}

[i][n][t] { return keyword\_int;}

[c][o][u][t] {return cout;}

[a-zA-Z\_][a-zA-Z\_0-9]\* { yylval.name = yytext; return identifier;}

. { /\* ignore everything else \*/ }

%%

*/\**

*int main(int argc, char \*\*argv)*

*{*

*FILE \*file = fopen("testcode","r");*

*if(!file){*

*printf("[-] Can't open the file!\n");*

*return -1;*

*}*

*yyin=file;*

*while(yylex());*

*fclose(file);*

*printf("\n");*

*}\*/*

### < Output >