**Lab 11 : Symbol Table Creation**

**Exercise 1**

Write a program to implement symbol table

**Input**

A program stored in a file. Read all the variables

**Out put**

Symbol table

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Scope** |
|  |  |  |

**Source Code:**

#include<iostream>

#include<fstream>

#include<string>

using namespace std;

bool lookup(string n, int s);

void insert(string n, string t, int s);

struct token{

string nn,tt;

int ss;

};

token tok[7];

int i=0,k=0,s=0;

string inp[]={"int", "a", ";", "int", "a", ";",""}, n,t;

int main()

{

while(i<6)

{

if(inp[i]=="int")

{

t=inp[i];

i++;

cout<<"int found\n";

if(inp[i]=="a")

{

n=inp[i];

// s=i;

i++;

cout<<"name found\n";

if(lookup(n,s)==false)

{

insert(n,t,s);

cout<<"Token Inserted"<<endl;

}

else

cout<<"Duplicate"<<endl;

if(inp[i]==";")

{

i++;

}

}

}

}

// cout<<"name: "<<n<<"\n Type: "<<t<<"\nScope: "<<s<<endl;

cout<<" Name Type Scope\n"<<endl;

for(int k=0;k<2;k++){

cout<<k<<": "<<tok[k].nn<<" "<<tok[k].tt<<" "<<tok[k].ss<<endl;

}

return 0;

}

bool lookup(string n, int s)

{

bool flag=false;

for(int x=0;x<2;x++)

{

if(tok[x].nn==n)

{

cout<<"match found "<<tok[x].nn<<endl;

flag=true;

break;

}

}

return flag;

}

void insert(string n, string t, int s)

{

tok[k].nn=n;

tok[k].tt=t;

tok[k].ss=s;

k++;

}