CD LAB 5 Generate and populate appropriate Symbol Table

Bhushan Sonawane
BE E 66 (BATCH 3)

yacc.y

```
%{
#include <stdio.h>
#include <malloc.h>
#include <string.h>
char vartype[10];
struct STable{
char label[10];
char type[10];
int size;
int location;
struct STable* next;
}*head;


void yyerror(const char *st){}
%}

%union { int size; char *label;}
%token NL
%token <label> ID
%token <size> INT FLOAT CHAR DOUBLE
%type <size> Type List
%%

S: S Declare
  | Declare
   ;

Declare: Type List ';'
   ;

Type: INT {strcpy(vartype ,"INT32"); }
   | FLOAT {strcpy(vartype ,"FLOAT32"); }
   | CHAR {strcpy(vartype ,"CHAR");}
   | DOUBLE {strcpy(vartype ,"DOUBLE64");}
   ;

List : List ',' ID { newSYM($3,vartype);}
   | ID { newSYM($1,vartype);}
   ;

%%
void DisplaySTable(struct STable*);

int main(){
stdin = fopen("in","r");
```

CD LAB 5 Generate and populate appropriate Symbol Table

```
freopen("out","w",stdout);
printf("%s\n","PARSING.....");
yyparse();
DisplaySTable(head);
}
int getVarSize(char* st){
if(strcmp(st,"INT32") == 0 ||strcmp(st,"FLOAT32") == 0  )
   return 4;
if(strcmp(st,"CHAR") == 0 )
   return 1;
if(strcmp(st,"DOUBLE64") == 0  )
   return 8;
}
void newSYM(char* lab, char* vartype){
struct STable *tnode = head;
int size = getVarSize(vartype);
if( !tnode ){
   struct STable* nnode = (struct STable *)malloc(sizeof(struct STable));
   strcpy(nnode->label ,lab);
   strcpy(nnode->type ,vartype);
   nnode->size = size;
   nnode->location = 100;
   nnode->next=NULL;
   head = nnode;

}else{
   while(tnode->next){
      if(strcmp (tnode->label,lab) == 0){
         printf("\nError: ReDeclaration of %s Variable %s (Previous Declaration as %s)",vartype,
lab,tnode->type);
         return;
      }
      tnode = tnode->next;
   }
   struct STable* nnode = (struct STable *)malloc(sizeof(struct STable));
   strcpy(nnode->label ,lab);
   nnode->size = size;
   strcpy(nnode->type ,vartype);
   nnode->location = tnode -> location + size;
   nnode->next=NULL;
   tnode->next = nnode;

   //return 0;
}
}

void DisplaySTable(struct STable *st){
int i = 1;
printf("\n\n\t\t\t%s\n","SYMBOL TABLE");
printf("\t| %s | Label | size | location |\n","Index");
```

CD LAB 5 Generate and populate appropriate Symbol Table

```
while(st->next){
   printf("\t|%7d|%7s|%6d|%10d|\n",i++,st->label,st->size,st->location);
   st = st->next;
}
}
```

lex.l

```
%{
#include <stdio.h>
#include "y.tab.h"
%}

letter [a-zA-Z]
digit [0-9]

%%
"int" {return INT;}
"float" {return FLOAT;}
"double" {return DOUBLE;}
"char" {return CHAR;}
{letter}({letter}|{digit})* { yylval.label = yytext; return ID;}
","|";" {return yytext[0];}
\n
%%
```

INPUT

```
int bh,sa;
char as,bh,sd;
double d4,as,sa;
float ff;
```

OUTPUT

```
Error: ReDeclaration of CHAR Variable bh (Previous Declaration as INT32)
Error: ReDeclaration of DOUBLE64 Variable as (Previous Declaration as CHAR)
Error: ReDeclaration of DOUBLE64 Variable sa (Previous Declaration as INT32)
```

SYMBOL TABLE

Index	Label	size	location
1	bh	4	100
2	sa	4	104
3	as	1	105
4	sd	1	106
5	d4	8	114
6	ff	4	118