<<<< USERNAME:tt FILENAME: maslabel.cpp >>>>

```
//---------------------------------------------------------------------
//File maslabel.cpp implements class Label
//---------------------------------------------------------------------
//Author: Thomas R. Turner
//E-Mail: trturner@uco.edu
//Date:   October, 2015
//---------------------------------------------------------------------
//Copyright October, 2015 by Thomas R. Turner.
//Do not reproduce without permission from Thomas R. Turner.
//---------------------------------------------------------------------
//C++ Standard include files
//---------------------------------------------------------------------
#include <iostream>
#include <fstream>
#include <iomanip>
#include <string>
#include <cstring>
#include <cstdlib>
using namespace std;
//---------------------------------------------------------------------
//Application include files
//---------------------------------------------------------------------
#include "maslist.h"
#include "maslabel.h"
//---------------------------------------------------------------------
//---------------------------------------------------------------------
struct LabelException {
    LabelException(const char* m)
    {   cout << endl;
        cout << "I am the Label Table and I am " << m << ".";
    }
};
//---------------------------------------------------------------------
//class Label resolves labels to addresses
//---------------------------------------------------------------------
Label::Label(int sz):size(sz),count(0)
{   D=new Descriptor[100];
    for (int a=0;a<size;a++) {
        D[a].defined=false;
        D[a].label="";
        D[a].address=0;
        D[a].L=0;
    }
}
//---------------------------------------------------------------------
//---------------------------------------------------------------------
Label::~Label(){if (D) delete[] D;}
//---------------------------------------------------------------------
//---------------------------------------------------------------------
int Label::Find(string l)
{   for (int a=0;a<count;a++) if (l==D[a].label) return a;
    return -1;
}
//---------------------------------------------------------------------
//---------------------------------------------------------------------
bool Label::IsFull(void){return count>=size;}
//---------------------------------------------------------------------
//---------------------------------------------------------------------
void Label::Define(string l,unsigned short adr,unsigned short M[])
{   int index=Find(l);
    if (index>0) {
        D[index].defined=true;
        D[index].address=adr;
        if (D[index].L) {
            for (D[index].L->First()
```

<<<< USERNAME:tt FILENAME: maslabel.cpp >>>>

```
                ;D[index].L->Next()
                )
            M[D[index].L->Element()]|=adr;
        }

    } else {
        DefinedInsert(l,adr);
    }
}
//---------------------------------------------------------------------
//---------------------------------------------------------------------
void Title(ostream& o)
{   o << endl;
    o << right << setw(15) << "Label";
    o << " ";
    o <<           setw(07) << "Defined";
    o << " ";
    o <<           setw(07) << "Address";
    o << " ";
    o << "References";
}
//---------------------------------------------------------------------
//---------------------------------------------------------------------
void RowPrint(ostream & o,string l,bool d,unsigned short a,List* L)
{   o << endl;
    o << right << setw(15) << l;
    o << " ";
    o << boolalpha << setw(07) << d;
    o << " ";
    o << "   ";
    o << setfill('0') << hex << setw(4) << a;
    o << setfill(' ') << dec;

}
//---------------------------------------------------------------------
//---------------------------------------------------------------------
void Label::Print(ostream& o)
{   Title(o);
    for (int a=0;a<count;a++) {
        RowPrint(o,D[a].label,D[a].defined,D[a].address,D[a].L);
    }
    o << endl;
}
//---------------------------------------------------------------------
//---------------------------------------------------------------------
int Label::Reference(string l,unsigned short a)
{   int index=Find(l);
    if (index>=0) {
        if (D[index].defined) return D[index].address;
        if (!D[index].L) D[index].L = new List;
        D[index].L->Insert(a);
    } else {
        UndefinedInsert(l,a);
    }
    return 0;

}
//---------------------------------------------------------------------
//---------------------------------------------------------------------
void Label::UndefinedInsert(string l,unsigned short a)
{   if (IsFull()) throw LabelException("full");
    D[count].defined=false;
    D[count].label=l;
    D[count].address=0;
    D[count].L=new List;
```

<<<< USERNAME:tt FILENAME: maslabel.cpp >>>>

```
    count++;
}
//---------------------------------------------------------------------------
//---------------------------------------------------------------------------
void Label::DefinedInsert(string l,unsigned short a)
{   if (IsFull()) throw LabelException("full");
    D[count].defined=true;
    D[count].label=l;
    D[count].address=a;
    D[count].L=0;
    count++;
}
```