

Compiling and Linking Multiple Files

- Modularity principles are applied in industry, by dividing a large project into sub-modules which can be coded by several programmers.
- C++ Functions are used to create these sub-modules.
- Each programmer can encapsulate all the functions that they have coded into a single file.
- These files are linked to the main program using a single project file and one or more header files.
- The header files contain function prototypes for all functions called by the main program.

Copyright © 2001 R.M. Laurie 1

```
#include <iostream>
#include <iomanip>
#include <cstdlib>
#include <ctime>

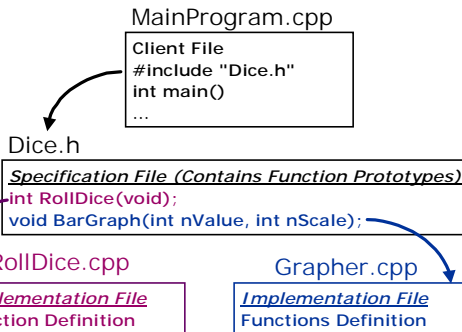
int RollDice(void);
void BarGraph(int nValue, int nScale);
using namespace std;
int main(void)
{
    int nI, nDiceCount[13];
    for(nI = 0; nI <= 12; nI++) nDiceCount[nI] = 0;
    srand((unsigned) time(NULL));
    for(nI = 1; nI <= 10000; nI++)
        nDiceCount[RollDice()]++;
    cout << "The distribution for 10,000 rolls of two dice is:\n";
    for(nI = 12; nI >= 2; nI--)
    {
        cout << endl << setw(2) << nI << " = " << setw(4)
            << nDiceCount[nI] << " |";
        BarGraph(nDiceCount[nI], 100);
    }
    cout << endl;
    return(0);
}

int RollDice(void)
{
    int nSum;
    nSum = 1 + (rand() % 6);
    nSum += 1 + (rand() % 6);
    return(nSum);
}

void BarGraph(int nCount, int nScale)
{
    int nI;
    for(nI = nCount/nScale; nI > 0; nI--) cout << "=";
```

Copyright © 2001 R.M. Laurie 2

Compiling and Linking Multiple Files



Copyright © 2001 R.M. Laurie 3

MainProgram.cpp File

```
#include <iostream>
#include <iomanip>
#include <cstdlib>
#include <ctime>
#include "Dice.h"
using namespace std;
int main(void)
{
    int nI, nDiceCount[13];
    for(nI = 0; nI <= 12; nI++) nDiceCount[nI] = 0;
    srand((unsigned) time(NULL));
    for(nI = 1; nI <= 10000; nI++)
        nDiceCount[RollDice()]++;
    cout << "The distribution for 10,000 rolls of two dice is:\n";
    for(nI = 12; nI >= 2; nI--)
    {
        cout << endl << setw(2) << nI << " = " << setw(4)
            << nDiceCount[nI] << " |";
        BarGraph(nDiceCount[nI], 100);
    }
    cout << endl;
    return(0);
}
```

Copyright © 2001 R.M. Laurie 4

Dice.h RollDice.cpp Grapher.cpp Files

```
int RollDice(void);
void BarGraph(int nValue, int nScale);

#include <cstdlib>
int RollDice(void)
{
    int nSum;
    nSum = 1 + (rand() % 6);
    nSum += 1 + (rand() % 6);
    return(nSum);
}

#include <iostream>
using namespace std;
void BarGraph(int nCount, int nScale)
{
    int nI;
    for(nI = nCount/nScale; nI > 0; nI--) cout << "=";
```

Copyright © 2001 R.M. Laurie 5

Visual C++ Multiple Files

- Create the source .cpp and .h files and save to same folder
- Close Workspace
- Create new project by going to **File | New**
- Select Projects Tab and Win32 Console Appl.
- Select the Location, create a project name, and click OK. Then select Empty Project.
- These files are linked to the main program using a single project file and one or more header files.
- Individual Files are then added to the project by the menu **Project | Add to Project | Files**
- Select all files for project and **Build/Run to test**

Copyright © 2001 R.M. Laurie 6

Visual C++ Multiple Files

The screenshot shows the Visual C++ IDE interface. The title bar reads 'Dice - Microsoft Visual C++ - [D:\...\MainProgram.cpp]'. The menu bar includes File, Edit, View, Insert, Project, Build, Tools, Window, and Help. The toolbar contains various icons for file operations. The Solution Explorer on the left shows a workspace named 'Dice' containing a project with the following files: Source Files (Grapher.cpp, MainProgram.cpp, RollDice.cpp), Header Files (Dice.h), and Resource Files. The main editor window displays the code for 'MainProgram.cpp', which includes headers for <iostream>, <conio.h>, <ctime>, and <cstdlib>, and uses the namespace std. The main function contains a loop for rolling dice and outputting the results.

Copyright © 2001 R.M. Laurie 7

Entry.cpp Example Function

```

#include <conio.h>
#include <cctype>
void GetNum(/*out*/ char szDigits[], /*in*/ int nMax)
{
    int nI;
    for(nI = 0; nI < nMax; nI++)
    {
        while(1)
        {
            szDigits[nI] = _getch();           //Is the input a digit?
            if(!isdigit(szDigits[nI]))
            {
                _putch(szDigits[nI]);
                break;
            }
            else if((szDigits[nI] == '\r') || (szDigits[nI] == '\b'))
            {
                //Is the input a CR?
                //Is the input a ESC?
                for(; nI > 0; nI--)
                {
                    _putch('\b');           //Output Backspaces
                    _putch(' ');           //Ring Bell
                    _putch('\b');           //Output Backspaces
                }
                nI = 0;
            }
            else _putch('a');               //Ring Bell
        }
        if(szDigits[nI] == '\r') break;     //Is the input a CR?
    }
    szDigits[nI] = 0x00;
}
    
```

Copyright © 2001 R.M. Laurie 8

Test.cpp Main Function & Header File

Input.h

```
void GetNum(/*out*/ char szDigits[], /*in*/ int nMax);
```

Test.cpp

```

#include <iostream>
#include "Input.h"
using namespace std;

int main()
{
    char szCashierID[5];
    GetNum(szCashierID, 4);
    cout << endl << szCashierID << endl;
    return 0;
}
    
```

The screenshot shows the Visual C++ IDE interface for a project named 'DigitEntry'. The Solution Explorer on the left shows a workspace named 'DigitEntry' containing a project with the following files: Source Files (Entry.CPP, Test.CPP), Header Files (Input.h), and Resource Files. The main editor window displays the code for 'Test.cpp', which includes <iostream> and 'Input.h', and uses the namespace std. The main function calls GetNum to get a cashier ID and prints it.

Copyright © 2001 R.M. Laurie 9

Answer.cpp Example Function

```

#include <conio.h>
#include <cctype>
bool GetAnswer(char cTrue, char cFalse)
{
    char cEntry;
    while(1)
    {
        cEntry = toupper(_getche());
        if(cEntry == cTrue || cEntry == cFalse)
            break;
        else
        {
            _putch('\a');
            _putch('\b');
        }
    }
    if(cEntry == cTrue) return(true);
    else return(false);
}
    
```

Copyright © 2001 R.M. Laurie 10

Gambler Structure Example

```

struct Gambler
{
    char        szSSN[15];
    char        szWonLoss[7];
    int         nChipValue;
    int         nBought;
    char        szUSCitizen[4];
    char        szOldFart[4];
    int         nSalary;
    double      dfWon;
    double      dfTax;
    double      dfCashOut;
};
    
```

Copyright © 2001 R.M. Laurie 11

Passing Output File by Reference

Function Definition

```
void PrintResults(/* In */ Gambler& sResults,
                /* In*/ ofstream& ofileResults, /* In */ int nCustNum)
{
    ofileResults << "CUSTOMER " << nCustNum << " SUMMARY: " <<
    sResults.szWonLoss << endl;
    ...
}
    
```

Function Prototype

```
void PrintResults(/* In */ Gambler& sResults,
                /* In*/ ofstream& ofileResults, /* In */ int nCustNum);
```

Function Call

```
PrintResults(sGambler, ofileXaction, nI);
```

Copyright © 2001 R.M. Laurie 12