

Lab1: Introduction Visual C++

Objective

The objective of this lab is to teach students:

- To work with the Microsoft Visual C++ 6.0 environment (referred to as VC++).
- C++ program structure and basic input and output statements.
- To type, edit, compile, link, and execute basic C++ programs.
- To identify errors (syntax, runtime, and logic errors) in C++ and correct (debug) them.
- To add internal documentation (comments)
 - At the top of the code (a header) which contains student identification (name, person #, etc.) and information about the project (project name, purpose, etc.). A template for this information is given below.
 - Inside the code which explains the program statements.
- To bundle the programs and submit the lab work online.

Description

For this lab, you will type in, edit, compile, link, execute, and debug three (3) C++ programs in the VC++ environment. You will be given the code for each program. Each program will contain at least one error which you need to identify and correct. You will be graded on the identification of each error and the correct execution of the resulting code. Three types of errors have been introduced:

1. Syntax errors (e.g., typos, grammar mistakes)
2. Run-time errors (e.g. data errors, arithmetic errors)
3. Logic errors (e.g., algorithmic errors, design errors)

Evaluation

There are three (3) parts to this lab:

1. Welcome Program (25%)
2. Sharing Pizza Program (30%)
3. Number Guessing Program (45%)

Lab Setup

If you are working in Furnas 211, you will have to log into the Ubiquity environment and start VC++ to work on the programs. The TAs will show you how to do this during your first lab session. You may also work on your own personal computer or other computers around the campus that support VC++. The software is free at UB Micro. For details on how to install it, visit UB Micro in the UB Commons, or visit their website: www.ubmicro.com.

Visual C++ Environment

A **project** in the VC++ environment is a program. A **workspace** is a folder in which all project-related information is stored. When you create a project you may create it in a new workspace or add it to an existing workspace. Refer to the VC++ textbook for more details. Do the following:

1. Create a workspace called Lab1.
2. Create a project for each program and add it to the lab1 workspace. Create project names as follows:
 - a. Welcome1
 - b. SharingPizza1
 - c. NumberGuessing1

Template for Program Header

Place the following code at the beginning of every source code file that you submit for this class.

```

/*****
* NAME : your name *
* PERSON NUMBER : your person number *
* PROGRAM : Lab name *
* PURPOSE : 1-2 line summary of the purpose of the lab *
* DATE : Date of last update *
* PLATFORM : Microsoft Visual C++ 6.0 Pro *
* Known bugs: *
*****/

```

On-line submission of your code

All source code (.cpp and .h) and data files (.in and .out) created by you and generated by the programs that you implement will need to be submitted using the on-line submit_eas230a or submit_eas230b commands. Instructions on how to use these commands are available on the course website, and will be discussed in class and in lab.

Part 1: Welcome Program

25%

This program is very basic. It asks the user to input their name and then it a message to them.

Evaluation

This program is worth 25% of the grade. The evaluation is as follows:

- | | |
|--|-----|
| 1. What were the errors (describe them)? | 5% |
| 2. What kind of errors were they (syntax, run-time, and/or logic)? | 5% |
| 3. What lines were they on? | 5% |
| 4. Correct the problems and submit the working code | 10% |

For 1, 2, and 3, put the comments at the **END** of your program inside a large multi-line comment. For example:

```

/*
Error Description: missing semi-colon
Error Type: syntax
Line number: 2
*/

```

If there are multiple errors, repeat the above lines of code for each error.

Code

Create a new project called **Welcome** and add it to the workspace **Lab1**. Type it in exactly as it is given and save it as **Welcome1.cpp**. The code for this program is as follows:

```

#include <iostream>
#include <string>
using namespace std;

int main()
{
    string name
    cout << "Please enter your name:\t";
    cin >> name;
    cout << name << ", " << "welcome to EAS230!" << endl;
    return 0;
}

```

Program 2: Sharing Pizza Program

30%

This program is slightly more complex than the Welcome program as it uses variables and queries the user for input. It calculates the percent error between the expected and actual results and displays it to the user. Run the program with the following two data sets:

1. totalPizza = 10, totalPeople = 6 and peopleNotHungry = 1
2. totalPizza = 10, totalPeople = 6 and peopleNotHungry = 6

Evaluation

This program is worth 30% of the grade. The evaluation is as follows:

- | | |
|--|-----|
| 1. What were the errors (describe them)? | 5% |
| 2. What kind of errors were they (syntax, run-time, and/or logic)? | 5% |
| 3. What lines were they on? | 5% |
| 4. Correct the problems and submit the working code | 15% |

For 1, 2, and 3, put the comments at the **END** of your program inside a large multi-line comment. For example:

```
/*
Error Description:  missing semi-colon
Error Type:  syntax
Line number:  2
*/
```

If there are multiple errors, repeat the above lines of code for each error.

Code

Create a new project called **SharingPizza** and add it to the workspace **Lab1**. Type it in exactly as it is given and save it as **SharingPizza1.cpp**. The code for this program is as follows:

```
#include <iostream.h>

int main()
{
    double totalPizza, peopleNotHungry, totalPeople;

    cout << "Enter the number of slices of pizza remaining:\t";
    cin >> totalPizza;

    cout << "Enter the total number of people eating:\t\t";
    cin >> totalPeople;

    cout << "Enter the number of people who are not hungry:\t";
    cin >> peopleNotHungry;

    cout << "The number of pieces of pizza that each hungry person can get is:\t"
        << totalPizza / (totalPeople - peopleNotHungry) << endl;
    return 0;
}
```

Program 3: The Number Guessing Program

45%

This program is more complex than the sharing pizza program as it uses flow control structures.

Evaluation

This program is worth 45% of the grade. The evaluation is as follows:

- | | |
|--|-----|
| 1. What were the errors (describe them)? | 5% |
| 2. What kind of errors were they (syntax, run-time, and/or logic)? | 5% |
| 3. What lines were they on? | 5% |
| 4. Correct the problems and submit the working code | 20% |
| 5. Demonstrate compiling, linking, and executing the program | 10% |

For 1, 2, and 3, put the comments at the **END** of your program inside a large multi-line comment. For example:

```
/*
Error Description:  missing semi-colon
Error Type:  syntax
Line number:  2
*/
```

If there are multiple errors, repeat the above lines of code for each error.

Code

Create a new project called **NumberGuessing** and add it to the workspace **Lab1**. Type it in exactly as it is given and save it as **NumberGuessing1.cpp**. The code for this program is as follows:

```
#include <iostream.h>
#include <time.h>
#include <stdlib.h>

int main(
{
    enum Status { CONTINUE, BIGGER, SMALLER, OVER };
    int guessTimes(0), myNumber, guessNumber;
    Status gameStatus(CONTINUE);

    srand( (unsigned)time( NULL ) );          // set myNumber randomly
    myNumber = (int) (((double)rand() / RAND_MAX) * 100);
    cout << "I've chosen a integer between 0 and 100.\n"
         << "Try to guess it! You have only 7 chances!" << endl;

    cout << "\nWhat's your guess?\t";        // first guessing
    cin >> guessNumber;
    guessTimes++;

    if (guessNumber == myNumber)
    {
        gameStatus = OVER;
        cout << "That's it! Well done!\n" << endl;
    }
    else if (guessNumber > myNumber)
    {
        gameStatus = BIGGER;
        cout << "Too high ... " << endl;
    }
}
```

```

else if (guessNumber < myNumber)
{
    gameStatus = SMALLER;
    cout << "Too low ... " << endl;
}

while ( gameStatus != OVER )    // keep guessing
{
    cout << "\nNext guess:\t";
    cin >> guessNumber;
    guessTimes++;

    if (guessNumber == myNumber)
    {
        gameStatus = OVER;
        cout << "That's it! Well done!\n" << endl;
    }
    else if (guessTimes >= 7)
    {
        gameStatus = OVER;
        cout << "You've tried 7 times. Sorry, you lost!" << endl;
        cout << "The answer was: " << myNumber << endl;
    }
    else if (guessNumber < myNumber)
    {
        gameStatus = BIGGER;
        cout << "Too high ... " << endl;
    }
    else if (guessNumber > myNumber)
    {
        gameStatus = SMALLER;
        cout << "Too low ... " << endl;
    }
}
return 0;
}

```

```

/*****
* (C) Copyright 2000 by Deitel & Associates, Inc. and Prentice Hall. *
* All Rights Reserved. *
* *
* DISCLAIMER: The authors and publisher of this book have used their *
* best efforts in preparing the book. These efforts include the *
* development, research, and testing of the theories and programs *
* to determine their effectiveness. The authors and publisher make *
* no warranty of any kind, expressed or implied, with regard to these *
* programs or to the documentation contained in these books. The *
* authors and publisher shall not be liable in any event for *
* incidental or consequential damages in connection with, or arising *
* out of, the furnishing, performance, or use of these programs. *
*****/

```