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. Hello World Program (25%)
2. Percent Error Program (30%)
3. Craps 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.

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 handout “Working with VC++ programs” 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. HelloWorld
   b. PercentError
   c. Craps
Template for Program Header
Place the following code at the beginning of every source code file that you submit for this class.

```cpp
/****************************
* 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) will need to be submitted using the on-line command which will be given to you during the lecture and lab. The procedure for submission will be discussed in future lectures and labs.

Part 1: Hello World Program 25%
This program is the most basic program that can be written. All it does is print out a single line to the screen and exits.

Evaluation
This program is worth 25% of the grade. The evaluation is as follows:
1. What were the errors? 5%
2. What kind of errors were they? 5%
3. What lines were they on? 5%
4. Correct the problems and comment on the solution on the line preceding the error and submit the working code 10%

Code
Create a new project called **HelloWorld** and add it to the workspace **lab1**. Type it in exactly as it is given and save it as **HelloWorld.cpp**. The code for this program is as follows:

```cpp
#include <iostream.h>
int main()
{
    cout << "Hello World" << endl
    return 0;
}
```
Program 2: Percent Error Program

This program is slightly more complex than the Hello World 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. `expectedResult = 100` and `actualResult = 90`
2. `expectedResult = 0` and `actualResult = 10`

Evaluation

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

1. What was the error? 5%
2. What kind of error was it? 5%
3. What line was it on? 5%
4. Correct the problems and comment on the solution on the line preceding the error and submit the working code 15%

Code

Create a new project called `PercentError` and add it to the workspace `lab1`. Type it in exactly as it is given and save it as `PercentError.cpp`. The code for this program is as follows:

```cpp
#include <iostream.h>

int main()
{
    double expectedResult, actualResult;
    double percentError;

    cout << "Enter the Expected Value:\t";
    cin >> expectedResult;

    cout << "Enter the Actual Value:\t\t";
    cin >> actualResult;

    cout << "The Percent Error is:\t\t" << ((expectedResult - actualResult) / expectedResult) * 100 << " %\n"
    return 0;
}
```
Program 3: The Game of Craps Program 45%

This program is more complex than the Percent Error program as it uses flow control structures. The program allows a user to play the game of Craps.

Evaluation

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

1. What were the errors? 5%
2. What kinds of errors were they? 5%
3. What lines were they on? 5%
4. Correct the problems and comment on the solution on the line preceding the error and submit the working code 20%
5. Demonstrate compiling, linking, and executing the program 10%

Code

Create a new project called Craps and add it to the workspace lab1. Type it in exactly as it is given and save it as Craps.cpp. The code for this program is as follows:

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

int rollDice( void ); // function prototype

int main()
{
  enum Status { CONTINUE, WON, LOST };  // function prototype
  int sum, myPoint;
  Status gameStatus;
  char ch;

  srand( time( 0 ) );
  sum = rollDice();            // first roll of the dice
  switch ( sum ) {
  case 7:                  // win on first roll
    case 11: // win on first roll
      gameStatus = LOST;
      break;
    case 2:                 // lose on first roll
    case 3:                 // lose on first roll
    case 12:                 // lose on first roll
      gameStatus = WON;
      break;
    default:                 // remember point
      gameStatus = CONTINUE;
      myPoint = sum;
      cout << "Point is " << myPoint << endl;
      break;                 // optional
  }

  while ( gameStatus == CONTINUE ) {    // keep rolling
    cout << "Hit any character and Retrun to start a new roll \n";
    cin >> ch;
    sum = rollDice();
  }
}
```
if ( sum == myPoint )       // win by making point
    gameStatus = WON;
else
    if ( sum == 7 )       // lose by rolling 7
        gameStatus = LOST;

if ( gameStatus == WON )
    cout << "Player wins" << endl;
else
    cout << "Player loses" << endl;

return 0;

int rollDice( void )
{
    int die1, die2, workSum;

die1 = 1 + rand() % 6;
die2 = 1 + rand() % 6;
workSum = die1 - die2;
cout << "Player rolled " << die1 << " + " << die2
    << " = " << workSum << endl;
return workSum;
}