**Student Lab Activity**



1. Lab # CIS CIS170C-A1
2. Lab 1 of 7: Getting Started (Your First C++ Programs)

1. Lab Overview - Scenario/Summary

Welcome to Programming with C++. The purpose of this three-part lab is to walk you through the following tutorial to become familiar with the actions of compiling and executing a C++ program.

In general, this lab will instruct you on:

1. how to create a project;
2. how to enter and save a program;
3. how to compile and run a program;
4. how to, given a simple problem using input and output, code and test a program that meets the specifications; and
5. how to debug a simple program of any syntax and logic errors.
6. Deliverables

|  |  |  |
| --- | --- | --- |
| **Section** | **Deliverable** | **Points** |
| **Part A** | Step 6: Program Listing and Output | **15** |
| **Part B** | Program Listing and Output | **15** |
| **Part C** | Program Listing and Output | **15** |

1. Lab Steps

**Preparation:**

If you are using the Citrix remote lab, follow the login instructions located in the iLab tab in Course Home.

**Lab:**

|  |
| --- |
| **Part A: Getting Started** |
| **Step 1:** Start the Application |
| 1. From the File menu, choose "New Project." 2. Choose “Win32 Console Application.” 3. Enter a name in the name field. 4. Click “OK” 5. Click “Next” and choose the following options: 6. Application Type: "Console Application" 7. Additional options: Check mark “Empty project”. 8. Click Finish. Your project is now created. |
| **Step 2:** How to Add a Source Code File to Your Project (.cpp file) |
| 1. In the Solution Explorer, right-click on the “Source Files” folder and select "Add" and then "New Item." 2. In the next dialog box, choose C++ file (.cpp), enter a name for your source code file, and press the Add button. 3. Type or copy and paste your code into the newly created source code file. Build the file by pressing F7, and then execute your program by pressing CTRL-F5 (start without debugging) or use the F5 key (Start Debugging). |
| **Step 3:** Create a Source Code File |
| Now enter the following C++ program **exactly** as you see it. Use the tab where appropriate. [Note: C++ is case sensitive.] Instead of John Doe, type your name.  **#include <iostream>**    **using namespace std;**  **void main()**  **{**  **cout << "John Doe" << endl;**  **cout << "CIS170C - Programming using C++\n";**  **cout << "\n\n\nHello, world!\n\n";**    **}**  When you execute a program in debug mode (F5), the console screen may appear and disappear  before you have an opportunity to view your output. There are several techniques you can use to  pause the console screen so you can read the output. On the very last line in the main() function:  a. insert the statement: system("pause");  -OR-  b. insert an input statement: cin >> myVarable; |
| **Step 4:** Output |
| The black screen or console should read:  John Doe  CIS170C - Programming using C++ Hello, World  **-** |
| **Step 5:** Save Program |
| Save your program by clicking **File** on the menu bar and then clicking **Save Program.cpp**, or by clicking the **Save** button on the toolbar, or **Ctrl + S**. |
| **Step 6:** Build Solution |
| To compile the program, click **Build** on the menu bar and then click the **Build** **Solution** or **Build LabA** option. You should receive no error messages. If you see some error messages, check the code above to make sure you didn't key in something wrong. Once you make your corrections to the code, go ahead and click Build >> Build Solution again. |
| **Step 7:** Execute the Program |
| Once you have no syntax errors, to execute or run your program, click **Debug** on the menu bar, and then click **Start Without Debugging.** |
| **Step 8:** Capture the Output |
| Print a picture of your screen output. (Do a print screen and paste this into MS Word.) |
| **Step 9:** Print the Source Code |
| Copy your source code and paste it into the same Word document as your screen print. Save the Word Document as *Lab01A\_LastName\_FirstInitial*  Note: Using the Visual Studio editor to compile your programs creates a lot of overhead. These additional files will become important as you create more sophisticated C# projects. Projects can contain one or more source-code files. For this course, you will not have to worry about all the extra files that are created. |
| **End of Part A** |

|  |
| --- |
| **Part B: Calculate Total Tickets** |
| **Step 1:** Create New Project |
| Now create a new project and name it LAB1B. Make sure you close your previous program by clicking File >> Close Solution. |
| **Step 2:** Type in Program |
| Like before, enter the following program. Type in your name for Developer and current date for Date Written.  **// ---------------------------------------------------------------**  **// Programming Assignment: LAB1B**  **// Developer: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **// Date Written: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **// Purpose: Ticket Calculation Program**  **// ---------------------------------------------------------------**  **#include <iostream>**    **using namespace std;**  **void main()**  **{**  **int childTkts, adultTkts, totalTkts;**  **childTkts = 3;**  **adultTkts = 2;**  **totalTkts = childTkts + adultTkts;**  **cout << totalTkts << endl;**  **}** |
| **Step 3:** Save Program |
| Save your program by clicking **File** on the menu bar and then clicking **Save Program.cpp**, or by clicking the **Save** button on the toolbar, or **Ctrl + S**. |
| **Step 4:** Build Solution |
| To compile the program, click **Build** on the menu bar and then click the **Build** **Solution** or **Build LabB** option. You should receive no error messages. If you see some error messages, check the code above to make sure you didn't key in something wrong. Once you make your corrections to the code, go ahead and click Build >> Build Solution again. |
| **Step 5:** Execute the Program |
| Once you have no syntax errors, to execute or run your program, click **Debug** on the menu bar, and then click **Start Without Debugging.** |
| **Step 6:** Capture the Output |
| 1. Capture a screen print of your output. (Do a PRINT SCREEN and paste into an MS Word document.) 2. Copy your code and paste it into the same MS Word document that contains the screen print of your output. 3. Save the Word Document as Lab01B\_LastName\_FirstInitial. |
| **End of Part B** |

|  |
| --- |
| **Part C: Payroll Program** |
| **Step 1:** Create a New Project |
| Create a new project and name it LAB1C. Make sure you close your previous program by clicking File >> Close Solution.  Include a comment box like what you coded in Part B. This can go at the very top of your program. |
| **Step 2:** Processing Logic |
| You need to write a program that calculates and displays the take-home pay for a commissioned sales employee along with all of the deductions.  **Input**: Prompt the user for the weekly sales.  **Process**: Perform the calculations. The employee receives 7% of his or her total sales as his or her gross pay. His or her federal tax rate is 18%. He or she contributes 10% to his or her retirement program and 6% to Social Security.  **Output**: Display the results  **Sample Output from Lab 1: Enter Weekly Sales: 28000**  **Total Sales: 28000.00 Gross pay (7%): 1960.00**  **Federal tax paid:  352.80 Social security paid:  117.60 Retirement contribution: 196.00 Total deductions:  666.40  Take home pay: 1293.60 Press any key to continue . . .**  Flowchart: *(continued on next page)*    *Pseudo Code:*   1. *Declare variables* 2. *Accept Input - weeklySales* 3. *Calculate Gross Pay = Weekly Sales \* .07* 4. *Calculate Federal Tax = Gross Pay \* .18* 5. *Calculate Social Security = Gross Pay \* .06* 6. *Calculate Retirement = Gross Pay \* .1* 7. *Calculate Total Deductions = Federal Tax + Social Security + Retirement* 8. *Calculate Total Take Home Pay = Gross Pay - Total Deductions* 9. *Display the following on separate lines and format variables with $ and decimal.*    1. *Total Sales Amount: value of weekly sales*    2. *Gross Pay (.07): value of gross pay*    3. *Federal Tax paid (.18): value of federal tax*    4. *Social Security paid (.06): value of social security*    5. *Retirement contribution (.1): value of retirement*    6. *Total Deductions: value of total deductions*    7. *Take Home Pay: value of take home pay*   **Note: Use SetPrecisions(2) to format the output (see page 98 of the text). The statements should look something like the following:**  //include the iomanip header file at the top of the file  #include <iomanip>  //use fixed and setprecision(2) to format the number  //use setw(8) to control the width of the field  //use \t to control the spacing between fields  cout << fixed << setprecision(2);  cout << "Gross Pay (0.07):\t $" << setw(8) << grossPay << endl; |
| **Step 3:** Save Program |
| Save your program by clicking **File** on the menu bar and then clicking **Save Program.cpp**, or by clicking the **Save** button on the toolbar, or **Ctrl + S**. |
| **Step 4:** Build Solution |
| To compile the program, click **Build** on the menu bar and then click the **Build** **Solution** or **Build LabC** option. You should receive no error messages. If you see some error messages, check the code above to make sure you didn't key in something wrong. Once you make your corrections to the code, go ahead and click Build >> Build Solution again. |
| **Step 5:** Execute the Program |
| Once you have no syntax errors, to execute or run your program, click **Debug** on the menu bar, and then click **Start Without Debugging.** |
| **Step 6:** Capture the Output |
| 1. Capture a screen print of your output. (Do a PRINT SCREEN and paste into an MS Word document.) 2. Copy your code and paste it into the same MS Word document that contains the screen print of your output. 3. Save the Word Document as CIS170A\_Lab01\_C\_LastName\_FirstInitial. |
| **End of Part C** |
| **END OF LAB** |