SYDE 121 – Digital Computation - Lab #1

NOTE: ***You do not have to know how to program to complete this lab!***

The lab is designed for you to gain familiarity with the Dev-C++ programming environment, using the engineering undergraduate facilities. Although the lab is submitted, there will be no grade assigned to it. TAs will record that your submission was completed properly.

OVERVIEW

1. Log on to your Nexus account and follow the steps outlined in "Basic C++ Program".
2. Understanding basic program modifications and errors.
3. **BONUS TASK.** Advanced program modification.
4. Submit your lab to your LEARN dropbox.

TASK 1: Basic C++ Program

This is your guide to entering and running a rudimentary C++ program on NEXUS. Before you launch Dev-C++, use the Windows Explorer Program to create a directory on your N:\ drive called SYDE121. This is where you should save your C++ files, documentation files, etc. Avoid putting the SYDE121 directory under your "desktop" path - otherwise you will be plagued with desktop icons whenever you run a C++ program. Do not save your files on the D drive as it is a temporary drive.

Steps:

1. *Create directory*
   Under your SYDE121 directory, create another directory called “Lab01”. It is recommended that you open a new directory for each of your labs.

2. *Start Dev-C++*
   Go to the "Start" menu, then to “Programs”, then to "Programming", then to “Dev-C++” and finally to “Dev-C++”.

3. *Open Dev-C++ editor*
   From the **File** menu select **New** -> **Project**. Select “Empty project”. Do not worry about the other project types on the list. Type in a name for the project – use Lab0101 to represent the first lab and the first part of the lab. Make sure that the path is set to your Lab01 directory.

   You will see the Solution Explorer appear on the left hand side of the screen. This area allows you to navigate from one file to another. It should be empty for the moment.

4. *Adding a source file to the project.*
   Right-click on “Lab0101” in the panel on the left hand side. Select **New File**. An “[*] Untitled” window will appear in the right hand panel. Select **File** -> **Save As**, and name it “lab0101.cpp”.

5. *Edit your first C++ code*
   Now, type the C++ program listed below in your “lab0101.cpp” file. Don’t worry about understanding the syntax – this will be covered in the course over the next couple of lectures. We are just trying to learn how to enter code, compile it and execute it for the time being.

English translation	C++ program

Include comments about program	`//filename: lab0101.cpp`
	`// This program uses a loop`
	`// to calculate squares of`
	`// integers 3 to 9`

Include standard Input/Output functions	`#include <iostream>` `using namespace std;`
The main ( ) function is contained by { ...}	`int main( )` `{`
Define an integer i and set to 0	`  int i = 0;`

FOR i starting at 3 and ending at 9, in steps of 1 print "i = " the integer value of i, and the square of i, then go to a new line.	`  for ( i = 3; i < 10; i++ )`
	`  {`
	`  cout << "i = " << i`
	`  << " squared = "`
	`  << i * i`
	`  << endl;`

When loop is done, comment displayed.	`  }` `  cout << "\nFun!\n";`
The 'system' command instructs Windows to keep the console open. Without it, the console window will close as soon as the program finishes executing.	`  system("pause");`
Return '0' to indicate that main( ) function is finished.	`  return 0;` `}`

6. *Save your file*
   After typing in the above code, save your file (Ctrl-S is the shortcut for this).

7. *Compile and running your program*
   Before running your program you have to compile it. For your information, and we will talk about this more in class on Friday compiling takes the C++ text that you entered and converts it to a format that the computer can understand (namely, machine code).

   To compile the program select **Execute** -> **Compile**. This step will produce an executable file in your project folder. In order to run the .exe you may select **Execute** -> **Run**, or simply double click it.

   Hint: you can ask the computer to compile and then run the compiled code all in one step by selecting **Execute** -> **Compile&Run**.

   If there is any syntax error (not logical error) in the program you will be notified. If the compile and execution were both successful, you will see the output of your program in a separate black screen (a 'DOS' screen). In this case, you should see an output similar to the next figure.

   There are ways to compile individual files and then build later – we will learn about these when we start to integrate multiple files into the same compile sequence (later!).

lab0101.cpp program output

```
i = 3 squared = 9
i = 4 squared = 16
i = 5 squared = 25
i = 6 squared = 36
i = 7 squared = 49
i = 8 squared = 64
i = 9 squared = 81

Fun!
Press any key continue…
```

TASK 2: Understanding Basic Program Modification and Errors

1. *Create a text file.*
   Outside of Dev-C++, create a text file on your computer, with the name "*lab0102.txt*". Keep the file open.

2. *Exploring program modifications*
   For each of the following, make the requested change, re-compile, and re-run. After re-running the code at each step, examine the output, and then explain the difference in the output you see now compared to the original output in the *lab0102.txt* file. Label each explanation as 2(a), 2(b), etc. For d-f, note the exact error message in the text file, along with your explanation of why the error occurred.
   a) Change `< 10` to `<=10` and run the program again.
   b) Remove `<< endl` from the `cout` (output line) and re-run.
   c) Put back `<< endl`, and change `i++` to `i+=2` and rerun.
   d) In the FOR loop, change `i++` to `i+` and re-run. Note the error message, then fix.
   e) Remove the semi-colon from the `int i = 0` line. Note the error message, then fix.
   f) In the `iostream` directive, add a space between the "`<`" the `iostream.`

3. *Explore Dev-C++*
   You are welcome to spend a few minutes exploring the other menus, function keys, and "Help". Be careful not to destroy the Lab0101 project in the process.

4. *Go back to your original "lab0101.cpp"*
   Change the statement "Fun!" to read "A program by: your first and last name". Re-run the program.

5. *Create a header.cpp*
   Open a new file by selecting **New** -> **Source File** from the **File** menu. Copy the following information in your header file. Replace 'XXX' with the correct information. Save your file as *header.cpp* for future use. Copy the text from header file and paste it at the top of your *lab0101.cpp* file.

header.cpp

```
//*****************************************
//  Student Name:  XXX
//  Student Number: XXX
//
//  SYDE 121    Lab: XXX  Assignment: XXX
//
//  Filename:  XXX
//  Date submitted: XXX
//
//  I hereby declare that this code, submitted
//  for credit for the course SYDE121, is a product
//  of my own efforts.  This coded solution has
//  not been plagiarized from other sources and
//  as not been knowingly plagiarized by others.
//
//*****************************************
```

6. *Create a README.txt text file*
   A common practice in the software industry is to include a file called *README.txt* whenever software programs are distributed. For SYDE 121, you will be required to submit a *README.txt* file for each of the programs you submit that provide detailed instructions on how to run the program. If your program does not take any input from the user, for example for the program in lab0101.cpp, the instructions will be as simple as will be as the following.

   ```
   Instructions to run the program:
   Step 1. Compile and run <yourfilename.cpp>.
   Step 2. Press any key to dismiss the window.
   ```

   If your program requires the user to provide some input, you need more detailed instructions. These instructions should clearly state what type of input is needed at each step, for instance, `Step X. Enter an integer and press Enter.`

   a) Open Windows WordPad (or your preferred text editor). Save the file as *README_Lab0101.txt* to your "Lab01" directory.

b) Add the header text in the *header.cpp* box above, replacing "XXX" with the correct information.
c) Below this header information add the text:

```
Instructions to run the program:
Step 1. Compile and run <yourfilename.cpp>.
Step 2. Press any key to dismiss the window.
```

d) Save the file.

TASK 3 BONUS TASK: Advanced Program Modification (Optional)

1. *Adding a source file to the project.*
   Right-click on "Lab0101" in the panel on the left hand side. Select **New File**. An "[*] Untitled" window will appear in the right hand panel. Select **File** -> **Save As**, and name it "*lab0103.cpp*".

2. *Copy your original program into this file.*
   Copy the code from your *lab0101.cpp* into the file *lab0103.cpp*.

3. *Make these changes to the program*
   a) In the program, create a function that checks if one number (**`int possibleSquareNum`**) is the square root of another number (**`int possibleRootNum`**).
   b) Create a second function that checks if one number (**`int possibleRootNum`**) is the square of another number (**`int possibleSquareNum`**).
   c) Now modify the main program to read in two numbers input by the user and then report (i.e., output) whether the first input number is the square, square root, or neither of the second number.

4. *Create a README_Lab0103.txt*
   Follow the basic instructions under 2.6 above for making a README file, but include more detailed instructions appropriate for this bonus task.

TASK 4. Submit your lab to your LEARN dropbox, in the format indicated below.

Use the following directions whenever you are submitting your lab code to your LEARN dropbox.

1. Login to your LEARN account.
2. Open the SYDE 121 course.
3. Under the "Dropbox", find the current Lab Assignment dropbox (for this lab it would be "Laboratory Assignment 1".
4. Upload the following files (one at a time).
   a) Your C++ source code (*.cpp), ***not*** the executable (*.exe), using the following format.

      Ensure that your source code file is named using the following format:

      - Filename begins with "lab" followed by
      - the lab number (two digits) followed by
      - the exercise number (two digits)
      - Example: lab0302.cpp (program for lab 3, question 02)

   b) Your *lab0102.txt* file.
   c) Your bonus C++ source code (*.cpp), if applicable.
   d) Your README file(s) (i.e. *README_Lab0101.txt and README_Lab0103.txt (if applicable)*)

Due Date

All material must be submitted to your LEARN dropbox by Friday, Sept. 20 by 1:30pm.

Note: Lab assignments are typically due the Friday afternoon after the lab session that week.