GE U111 Engineering Problem Solving and Computation - Spring 2004
Tutorial for using Microsoft Visual C++ on NUNET

- Bring a **3.5" high density disk** to the computer lab. Format it if it isn't already formatted.
- When entering the lab, **scan your disk for viruses.** If you write anything to your disk, you should also scan for viruses when leaving. This small step can save you and your fellow students untold aggravation.

1 ***DO NOTHING NOW - JUST READ THE CODE*** to see where you're headed. In this tutorial you will write a program that computes the volume of 2 triangular pyramids that are 7 cm high, but have differing base dimensions. The final code will look as follows.

```
//      Program to compute volume of two triangular pyramids that are 7 cm high
//      type your name here
#include <iostream>
#include <fstream>
using namespace std;
#define HEIGHT 7.
int main()
{
        ofstream outfile;
        outfile.open("a:PYRabc.out");
        double side_1, side_2, base_area, volume;
        side_1 = 0.6;
        side_2 = 3.0;
        base_area = side_1*side_2 / 2.;
        volume = base_area*HEIGHT / 3.;
        cout <<"First pyramid volume is " << volume << " cubic cm" << endl;
        outfile <<"First pyramid volume is " << volume << " cubic cm" << endl;
        outfile <<"Sides 1 and 2 are " << side_1 << " and " <<side_2 << " cm " << endl;

        cout <<"Enter new lengths for sides 1 and 2:"<< endl;
        cin >> side_1 >> side_2;
        base_area = side_1*side_2 / 2.;
        volume = base_area*HEIGHT / 3.;
        cout << "Second pyramid volume is " << volume << " cubic cm"<<endl;
        outfile <<"Second pyramid volume is " << volume << " cubic cm" << endl;
        outfile <<"Sides 1 and 2 are " << side_1 << " and " <<side_2 << " cm" << endl;
        outfile.close();
        return 0;
}
```

2 **Access Microsoft Visual C++.** Select, in order, *Start, NUNET, Applications, Programming, Microsoft Visual Studio, Microsoft Visual C++*. Visual C++ includes software for writing programs in two related languages: C and C++. In GE U111, we'll be using the language C ++.

3 **Create a new file.** Select ***File / New.*** Select the ***Files*** tab, and ***C++ Source File***. In the text boxes on the right, enter

- file name ***PYRabc.cpp*** where *abc* is your initials. File names are *not* case sensitive.
- Location – use the default, which should be the hard drive (either C: or M:). ***In the labs, use the default folder, M:\Personal***. Later, you'll have to copy your source code from this location onto your floppy disk.
- Click ***OK***

The extension "***.cpp***" tells the compiler your program is written in the "C++" language. [If you don't give an extension, you will automatically give your program the extension "***.cpp***", which is for programs written in the "C++" language. You want the extension .CPP, not .C]

4 **Type the first part of the code.** Point and click in the `PYR`*abc*`.C` window. ***For comments use //***. The "C++" language is ***case sensitive***, so be sure to type the lowercase and UPPERCASE letters as shown. Use blank lines, tabs (for indentation), and spaces as they appear, but don't worry about copying the amount of "whitespace" exactly. Where you see ***italics*** you are supposed to replace it what is requested. You will see that Visual C++ does some indentation for you automatically.

5 **Save your work** by selecting ***File / Save***. It is important to get into the habit of saving your work periodically. If the system crashes or you mistakenly execute a command, you will be able to revert to your saved file. (For your further protection, Visual C++ also makes backup copies of your program.)

6 **Take advantage of Copy & Paste.**

(a) The next line of code (`outfile` ...) is almost identical to the last line you typed (`cout` ...). First use the Copy and Paste features of the editor to copy the last line. *(Please ask for help if you are unsure how to do this.)* Then edit the newly pasted line to make it match what the next line should be, by changing `cout` to `outfile`.

(b) [For your information, `cout` will cause information to be printed to the screen, which is helps the user interactively control the program. The `outfile` will cause information to be printed to an output file that can be saved and printed, but can't be seen while the program is

running.]

7 **Save your work** by selecting ***File / Save***.

8 **Finish entering the code and save it.** A lot of it can be done by copying & pasting to make it quicker. Check that your code matches the code given at the beginning of this tutorial; if not, edit it and save again.

9 **Compile the code** by selecting ***Build / Compile***. Several things will happen

- You'll be asked if you want to open a default workspace. Answer Yes.
- A workspace window appears on the left of the screen. It offers two views: the Class View, which we won't care about for now, and the File View, which will list the various files you have open in your project (the software has created a Project for you!). Click on the File View icon at the bottom of the workspace window.
- A message window appears on the bottom. If there were no errors in your program, it will conclude with

  ```
  pyrabc.obj - 0 errors, 0 warnings.
  ```

  [The compile step involves creation of a new file, called the object file. If you search for it in your computer, you can find it. But you can't read it or do anything else with it – it's in machine code.]
- If there were errors, correct them. First, you might want to expand the message window *(ask how if you need help)* and scroll up till you see the error messages. Then double click on the first error message, and a little arrow will appear in the source code window telling you where the error is. *Note: the computer is not perfect (but pretty good) in pointing you to the exact location of the error. If the error is not in the line to which you were pointed, work back "up" your code, line-by-line, until you find the error* . Find the error and fix it. *Note: one "human" error in your source code can cause multiple error messages, so fixing one "bug" may take care of several errors; you'll be able to tell by compiling again.* Repeat the process of finding the first error in the list, correcting it, and re-compiling till your program compiles successfully.

10 **Build the executable file** by selecting **Build / Build** or **Build / Rebuild All**. Messages will appear in the message window that the program is linking (FYI, it's linking to functions in the `stdio` library). The final message should be

```
pyrabc.exe - 0 error(s), 0 warning(s)
```

In the link / build step, an executable file (*.exe) is created. [This is your program completely translated from C to machine language and ready to run. Normally you'll run it from within Visual C++, but you could also run it from Windows by just double clicking on the file's icon – but save that for a later time.]

11 **Run the program – but don't close the output window yet.** Select ***Build / Execute***. A Console Window will appear. As the program runs, follows the prompts and enter numbers. When more than one number is requested, separate them by whitespace (a blank space, a tab, or a newline). For the first pyramid, side lengths of 0.6 and 3.0 cm were already specified in your code. For the second pyramid, enter your choice of side lengths. Before you let the Console Window close….

12 **View and close the Console Window.** Before you let the Console Window close, please note what appears. It should be exactly what the program contains in its `cout` statements, plus the values that you entered. Do your own hand calculation to confirm that the first pyramid result is correct. Note, the Console Window has a limited length. If a program sends a lot of output to the console, only the last 20 or so lines will remain visible.

The prompt, `"Press any key to continue:`, should appear on the screen -- it was automatically supplied by Microsoft Visual C++.

13 **View the output file.** Through its `outfile` statements, your program also printed some output to a file on your floppy disk named **A:PYR*abc*.OUT** (where *abc* are your initials and A: is the path to the floppy drive). You can open that file using the Visual C++ editor; just select ***File / Open*** and select the correct file. It will contain output from the `outfile` statements. Notice that it includes the base side lengths you either specified or chose for each pyramid.

```
First pyramid volume is 2.1 cubic cm
Sides 1 and 2 are 0.6 and 3.0 cm
Second pyramid volume is X cubic cm
Sides 1 and 2 are Y and Z cm          // Y and Z are whatever input values you entered
```

Confirm that the output for the second pyramid is correct.

14 **Print your output file and your source file.**

(a) To print the output file, be sure the cursor is somewhere in that file's window, then select ***File / Print.***

(b) Were you wondering if you can print the Console Window? On some systems, a "print screen" will do the trick. But, we'll always print from an output file that was filled by `outfile` statements.

(c) To print the source file (i.e., the program), first you've got to get it to appear. Select ***Window / Cascade*** (or ***Tile Horizontal***) and all your open files will appear, including the source code. Click somewhere in the source code's window, then select ***File / Print.*** Pick up your printed output (after a few moments) at the printer.

(d) For your information, the source file and the output file are both simply text files. You can view them, print them, format them, etc., with any text editor, including a word processor.

15 **Make a few common programming errors and see what happens.** You're likely to make errors like these, and you should become familiar with how the computer will react. Try these three, one at a time. Refer to Step 9 for how to view and respond to error messages during compiling and linking.

a. Error 1: Delete a semicolon (;) from the end of a line (any line). Try compiling and making sense of the error message that occurs.

b. Error 2: Delete angle brackets *(>>)* from one of the `cout` statements.

c. Error 3: Delete `#include <iostream>` from the program. You'll see that the program compiles OK, but fails to link because the linker doesn't recognize words that are defined in the *iostream* library header, like `cout` and `cin`.

16 **Exit Visual C++.** Pretend you're done for the day. Close the program. Be sure that you save a version without intentional errors!

17 **Copy your source code onto your floppy.** DON'T FORGET THIS STEP or you'll regret it! Ask if you need help.

18 **Return to Visual C++ to modify your program.** Imagine now that you're returning a day later; you want to change your program a little. First, copy your source code from your floppy onto the hard drive (M:\Personal in the computer labs). Get back into Visual C++. To open your program, select ***File / Open***. Under *Look in*, select the hard drive (not the floppy (A:)), and a listing of all the files with the requested extension will appear. Select the file you want by pointing and clicking; then hit **OK**. (Warning: you can open the file directly from A:, but if you try to build the executable program from A:, expect troubles.)

19 **Look at the new files you made!** Do a search in your computer (Start / Find / File) for files named PYR*abc*.* You'll see lots of files: your source file, the object file, the executable file, and several other files that have a role in advanced C++ programming, but are irrelevant for this course.

Don't forget to take your diskette and printouts with you when you leave!