Lab 2 - CMPS 1044, Computer Science I
Introduction to Visual Studio - File Input and Output

This lab will focus on reading input from a file and printing output to a file using the fstream library. This is different from last week's lab, which used the iostream library for printing output to the screen. First, let's review how to create a new project and add a source file.

To create a new project
1. From the **File** menu, point to **New**, and then click **Project**.
2. Click **Visual C++**. Then, click **Win32 Console Application**.
3. Type a name for the project.

   You can accept the default location or browse to a directory where you want to save the project, such as a flash drive.
   Press **OK** to start the **Win32 Application Wizard**.
4. On the **Overview** page of the **Win32 Application Wizard** dialog box, click **Next**.
5. On the **Application Settings** page under **Additional options** select the **Empty Project** setting and click **Finish**.

To add a new source file (program)
1. From the **Project** menu, click **Add New Item**.
   In the **Visual C++** area, select **Code**. Then click **C++ File (.cpp)**.
2. Type the Name of the file to be added and click **Add**.
3. In the editing window, type the code below for the program. Note: punctuation, spacing and indention are important! Blank lines are used for readability, but do not affect the execution of the program.

```cpp
#include <iostream>
#include <fstream>
using namespace std;
int main()
{       int birth_Date;

        ifstream infile;
        ofstream outfile;
        infile.open("input.txt");
        outfile.open("output.txt");

        infile >> birth_Date;
        // You may put other statements here
        outfile << "Your name here \n";
        outfile << birth_Date;

        infile.close();
        outfile.close();
        return 0;
}
```

4. Press **CTRL + F5** to compile and execute your program. Because all output was to a file, you will NOT see the Display Window with results on the screen. Observe the Output Window at the bottom of the screen. If no error messages appear, your program probably executed properly.

To View Your Results
1. From the **Project** menu, click **Add Existing Item**.
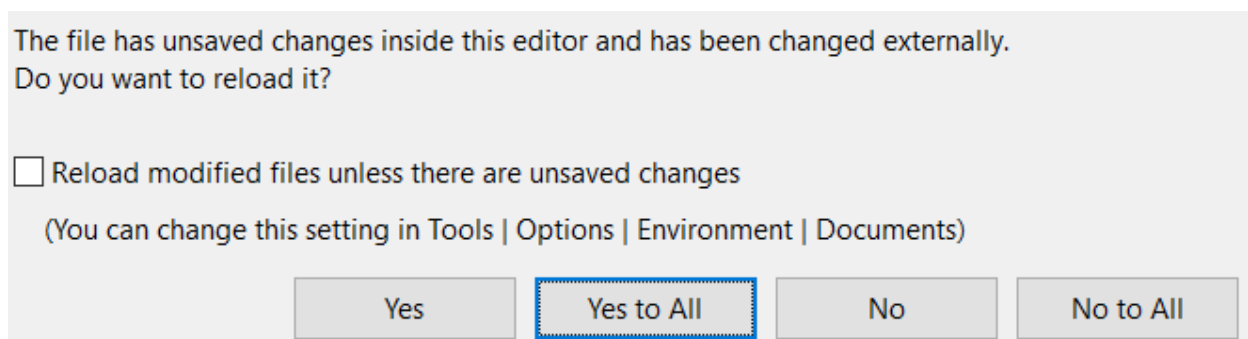2. Select the file named **output.txt**, then click **Add**.

What happened? The program compiled and executed without errors, but the `birth_Date` variable printed out junk. When the code is to read from an input file, we must place a file of that name in the same directory as our project.

To add a text file for input
1. From the **Project** menu, click **Add New Item**.
    In the **Visual C++** area, select **Utility**. Then click **Text File (.txt)**.
2. Type the Name of the file to be added (for example, type **input** if the file is to contain data to be read by the program) and click **Add**. The empty text file (**input.txt**) will be added to the project.

Type the year you were born into the **input.txt** editing window. If you wish not to say, type the year 1985. Then, compile and execute the program again by pressing **CTRL + F5**.
When Visual Studio asks if you want to make changes to **output.txt**, select **Yes** or **Yes to All**.

The file has unsaved changes inside this editor and has been changed externally.
Do you want to reload it?

☐ Reload modified files unless there are unsaved changes

(You can change this setting in Tools | Options | Environment | Documents)

| Yes | Yes to All | No | No to All |

Printing your Program Code
With your program code displayed on the screen, click the **File** menu in the upper left corner of the screen, then select **Print**. If you are using a color printer, your printout will be in color as on the screen, otherwise it will be B/W. Color is NOT necessary for work turned in to your instructors. Lab printers are not color.

Printing your Output files
With your file code displayed on the screen, click the **File** menu in the upper left corner of the screen, then select **Print**. Do this for each file that you want to print.

Documentation
Good programming practice insists that all program files <u>must</u> contain documentation. That is, comments from the programmer providing basic information about the program. Documentation lines begin with the **//** symbols. All programs should have at least the following documentation at the beginning of EVERY program. Your instructor may require additional documentation be included.

```
// Your first and last name
// Computer Science I - Instructor name
```

```
// Project # - Project Name
// Date
// Additional information as required by your instructor
```

In addition, **the body of your program should also be documented**. This includes adding comments for all components of your program. For more examples of appropriate documentation, see the sample programs in your textbook. Your lab assignment is fully documented. You are to **include at least 3 comments** as you complete the assignments **for every lab**.

Important notes: Sometimes you will want to type a program in Word or Notepad then copy into the Visual Studio programming window. If you do this, you MUST check it very carefully because there WILL BE errors in the copy. For example, quotes (" ") must be deleted and retyped after the copy; they will not be the correct version of quotes for C++. Word automatically capitalized some words. This will cause errors in C++, so you will have to retype some words, removing the capital letter at the first of the word or line. There are probably others.

Learn to recognize the colors in the Visual Studio editor; it will help you spot errors. For example, comments are always in green. Also, literals (words in quotes) will be in red. Commands are in black; reserved words in blue. Potential errors will have red squiggly lines underneath. Always write statements to print your full header to the output file for lab assignments.

If your code prints with text wrapping because your lines are too long (will mess up indentation and print an arrow symbol (↵)), edit your code so that each line is shorter than 80 characters, using the **COL** number at the bottom of Visual Studio as a guide.

LAB 2 – Assignment

Modify the program that you began in class to include the additional code below, including at least 3 comments. Create an input file containing your year of birth on the first line and the values 2.5 and 3.7 on the second line (as shown in the sample output below). Execute the program, correcting errors as necessary. Print off the program code and the output file. Staple them together (program code on top) and turn in under your professor's roll sheet.

```cpp
// Your Name
// Computer Science I – Instructor Name
// Lab 2 - File I/O
// Date
// This program opens an input file and creates an output file. It reads a
// year and two real numbers from the input file, performs arithmetic on the
// numbers, then prints the results to the output file.

#include <iostream>
#include <fstream>
using namespace std;
int main()
{
    // Variable declarations
    int birth_Date;             // declare integer variable for your birth year
    float A, B, Sum, Product;   // declare real variables for arithmetic
    ifstream infile;            // declare an input file
    ofstream outfile;           // declare an output file

    infile.open("input.txt");     // open "input.txt" for reading
    outfile.open("output.txt");   // create "output.txt" for writing
```

```cpp
        // Program Code for reading your year of birth
        infile >> birth_Date;   // read a value for birth year from input.txt

        // Program Code for performing some arithmetic operations
        infile >> A >> B;            // read values for A & B from input.txt
        Sum = A + B;                 // Add A & B, storing result in Sum
        Product = A * B;       // Multiply A & B, storing result in Product

        // Program Code to output results to the output file
        outfile << "your name here \n";     // write your name to "output.txt"
        // Write other output statements to print your full header to
        //"output.txt"

        outfile << birth_Date << "\n"       // write birth date to "output.txt"
        outfile << "A and B have the values " << A << " and " << B << "\n";
        outfile << "The sum is " << Sum << "\n";
        outfile << "The product is " << Product << "\n";

        // Commands to close files
        infile.close();   // Always close each file so that the
                          // I/O buffer will be
        outfile.close();  // written (otherwise, you may not get all of your
                          // results written to the output file)
        return 0;
}
```

Sample input & output files

****

Input File
1985
2.5 3.7

****

Output File
Your name here
Computer Science I...

1985
A and B have the values 2.5 and 3.7
The sum is 6.2
The product is 9.25