Style Guide

PART I

Why have style?

Why have style?

- Readability
- Reusability
- Modifiability
- Easier to debug!

You need to really understand these slides

Follow up with me if you have questions

Some style guidelines

Name identifiers properly
- Variables → lowercase
- Constants → UPPERCASE

Indent blocks of code

```
int main()
{
  indent here
}
```


Commenting your code

For all programs in this class

Before int Main
- Use comments to describe your program

Data Table
- The declaration section must contain a data table
- The data table
  - states the use of the variable or named constant &
  - how its value is obtained/used.

Other comments should be used throughout your code to
- Describe what each section is doing
  - (think in terms of input, processing, & output)
- Complicated parts of the code → be descriptive!

Data Tables

Should state: use of the identifier & how it is used

Comments should be lined up

All identifiers should have their own line and datatype

Which of these are correct?

```
int firstNum;          // INPUT - first value to average
int secondNum;         // INPUT - second value to average
float average;         // CALC & OUT  - average of two values
```

CORRECT

```
int firstNum; // INPUT - first value to average - INPUT
int secondNum; // INPUT - second value to average - INPUT
float average; // CALC & OUT  - average of two
```

INCORRECT

```
int firstNum;                  // input value
int secondNum;         // input value
float average;         // calculated average
```

INCORRECT

Create a Template

```
/*************************************************************
 *  AUTHOR       : Juan Leon
 *  Assignment #1: Template
 *  CLASS        : CS1B
 *  SECTION      : MW: 10:30a - 12p
 *  Due Date     : 1/5/12
 *************************************************************/
#include <iostream>
using namespace std;
/*************************************************************
 *
 * ADD TWO INTS
 *  ---------------------------------------------------------
 * This program accepts two integers in from a user, sums
 *   them and then outputs the result to the monitor.
 *  ---------------------------------------------------------
 * INPUT:
 *   inp1: First integer to be summed -> input from user
 *   inp2: Second integer to be summed -> input from user
 *
 * OUTPUT:
 *   sum:     The sum of the two ages
 *************************************************************/
int main()
{
    // constants
    int inp1;      // INPUT - First integer to sum
    int inp2;      // INPUT - Second integer to sum
    int sum;       // CALC & OUT - contains the result of
                   //     the sum of two inputs -

    // output the class heading to the screen
            cout << "****************************************\n";
    cout << "   Programmed by: Juan Leon";
    cout << "   Student ID   : 750125\n";
    cout << "   CS1B         : MW - 6p-7:30\n";
    cout << "   Lab #1       : Qt Tutorial\n";
    cout << "****************************************\n";
;

    // INPUT:  A description of what is being input.
    // PROCESSING:  Detail what is being processed.
    // OUTPUT:  Details of what is being output.
    return 0;
}
```

Class Heading

Pre-processor directives

General Program description

Data Table

Output Class Heading

Doc throughout code

Create a project

Put all this in there

Call it 0-template

Cut & paste the project

Class heading information

First lines in your source file

```
/*************************************************************
 *   AUTHOR     : Juan Leon
 *   Lab #1     : Template
 *   CLASS      : CS 002
 *   SECTION    : MW: 10:30a - 12p
 *   Due Date   : 1/5/12
 *************************************************************/
```

Note the alignment

Replace the data in purple with the appropriate data.

Next...

Preprocessor Directives then doc for the main program →
Including a list of inputs and outputs

```
#include <iostream>
#include<iomanip>
using namespace std;

/*************************************************************
*
* ADD & MULTIPLY TWO INTS
*
*_____________________________________________________________
* This program does whatever this program does
*  save this template and fill in the info appropriate
*  for your program
*_____________________________________________________________
* INPUT:
*   int1: First integer to be summed received as input
*   int2: Second integer to be summed received as input
*
* OUTPUT:
*   sum    : the sum of the two integers (int1 & int2)
*   product: The product
```

Program Title

General Description

Describe the Inputs & Outputs here

Notice the indentation

Next → int main

```
int main ()
{
        // Declare your constants here
        //       document constants above the declarations

        // Declare variables here - include your data table

        // Initialize variables

        // OUTPUT your header and class information here
        //    (see next slide)

        // INPUT:  A description of what is being input.

        // PROCESSING:  Detail what is being processed.

        // OUTPUT:  Details of what is being output.

        return 0;
}
```

Double space

Header & Class Information (omit for hypergrade submissions)

```
cout << "****************************************\n";
cout << "   Programmed by: Juan Leon\n";
cout << "   Student ID   : 750125\n";
cout << "   CS1B         : MW - 6p-7:30\n";

// put lab # or Assignment # as appropriate
cout << "   Lab # 7      : Lab Name\n";
cout << "****************************************\n";
```

It is easier if you show print margins

Click on

Tools->Option(PC)

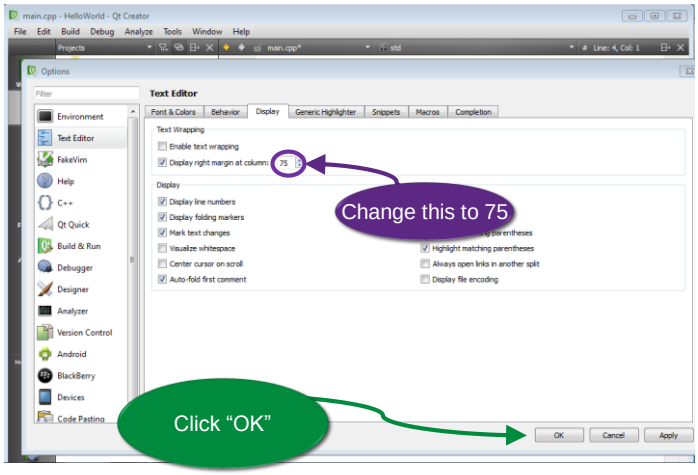
File->Preferences(Mac)

->Text Editor

->Display

1.Change

Display right margin at column to 75



Documenting executable code

```
int main()
{
     // Declare your constants here
     //       document constants above the declarations

     int num1;        // INPUT - first value to average
     int num1;        // INPUT - second value to average
     float  average;  // CALC & OUT - integer

     // INPUT -- get numbers to average from user
     cout  << "Enter first value to double: ";
     cin   >> num1;

     // PROCESSING -- calculate the average
     average = float(num1 + num2) / 2;

     // OUTPUT -- output the average
     cout  << "\n\nThis average is: " << average;
     return 0;

}

CORRECT
```

All programs have a data table

Space Between operators

Document above each code segment

Block of code is indented

Double space between code segments

Initializing Variables

DO NOT INITIALIZE VARIABLES IN THE DECLARATION SECTION.

Initialize variables just before their use in the program.

```
int count;
count = 0;
```

CORRECT

```
int count = 0;
```

INCORRECT