Matters of Style

- Why we have style?
- Guidelines
- Flowchart overview

Why have style?

- Why have style?
  - Readability
  - Reusability
  - Modifiability
  - Easier to debug!
- You need to really read this chapter
- 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 t
  - 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;        // IN & CALC    - first value to average
int    secondNum;       // IN & CALC    - second value to average
float average;          // CALC & OUT - average of two values
```

CORRECT

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

INCORRECT

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

INCORRECT

Create a Template

- Create a project
- Put all this in there
- Call it 0-template
- Cut & paste the project

Class Heading

Pre-processor directives

General Program description

Data Table

Output Class Heading

Doc throughout code

```
/*************************************************************
 *  AUTHOR       : Michele Rousseau
 *  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.
 *
 * ----------------------------------------------------------
 * INPUTS:
 *   inp1: First integer to be summed -> from user
 *   inp2: Second integer to be summed -> from user
 *
 * OUTPUT:
 *   sum:  The sum of the two inputs -> to the screem
 *************************************************************/
int main()
{
    // constants - include data table above for constants
    //             see the eclipse lab
    int inp1;      // IN & CALC  - First integer to sum
    int inp2;      // IN & CALC  - Second integer to sum
    int sum;       // CALC & OUT - contains the result
                   //              of the sum of two inputs

    // OUTPUT - class heading to the screen
    cout << left;
    cout << "****************************************\n";
    cout << "   PROGRAMMED BY: Michele Rousseau\n";
    cout << "   STUDENT ID   : 750125\n";
    cout << "   CS1B         : MW - 6p-7:30\n";
    cout << "   ASSIGNMENT #1: Template\n";
    cout << "****************************************\n";
    cout << right;

    // 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 information

First lines in your source file

```
/*************************************************************
 *   AUTHOR     : Michele Rousseau
 *   LAB #1     : Template
 *   CLASS      : CS1A
 *   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
*
* INPUTS:
*   int1: First integer to be summed -> from user
*   int2: Second integer to be summed -> from user
*
* OUTPUTS:
*   sum     : the sum of the two ages
*   product: The product of the two integers
*************************************************************/
```

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

```
// OUTPUT - class heading to screen
cout << left;
cout << "****************************************\n";
cout << "* PROGRAMMED BY : Michele Rousseau\n";
cout << "* STUDENT ID    : 750125\n";
cout << "* CS1A          : MW - 6p-7:30\n";

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

Change everything in purple to the appropriate information for the project. For assignments put "Assignment" instead of "Lab"

Or… just you're the code from your eclipse lab

It is easier if you show print margins

Right click on the scroll bar to the left of the Editor window to get this menu

1. Check show line numbers
2. Check Show print margins
3. Change Print margin column to 75

Documenting executable code

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

     int intNum;             // IN & CALC    - first value to average
     int intNumSquared;      // CALC & OUT - integer to store the doubled value

     // INPUT -- get numbers to average from user
     cout  << "Enter first integer to square: ";
     cin     >> intNum;

     // PROCESSING -- calculate the average
     intNumSquared = intNum * intNum;

     // OUTPUT -- output the average
     cout  << "\n\nThe integer squared is: " << intNumSquared;

     return 0;
}
```

All programs have a data table

Space Between operators

Document above each code segment

Double space between code segments

Block of code is indented

Initializing Variables

DO NOT INITIALIZE VARIABLES IN THE DECLARATION SECTION.

- Initialize variables just before their use in the program.

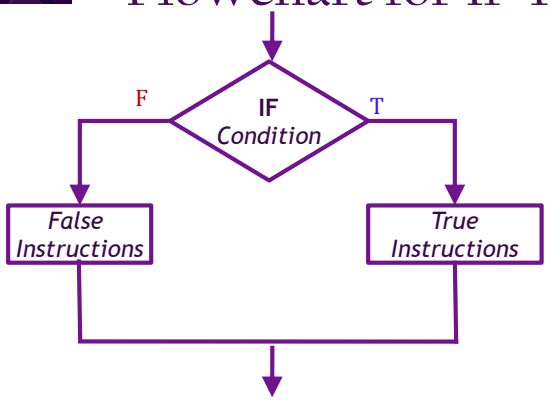
```
int count;

count = 0;
```

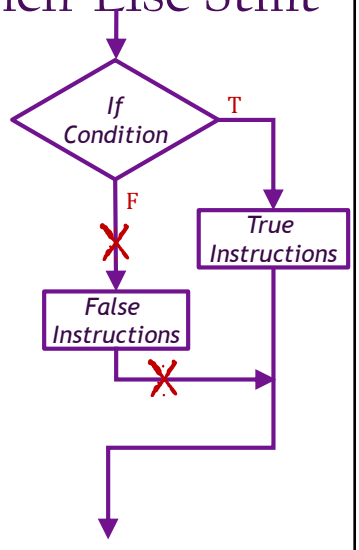
CORRECT

```
int count = 0;
```

INCORRECT

Flowchart for If-Then-Else Stmt

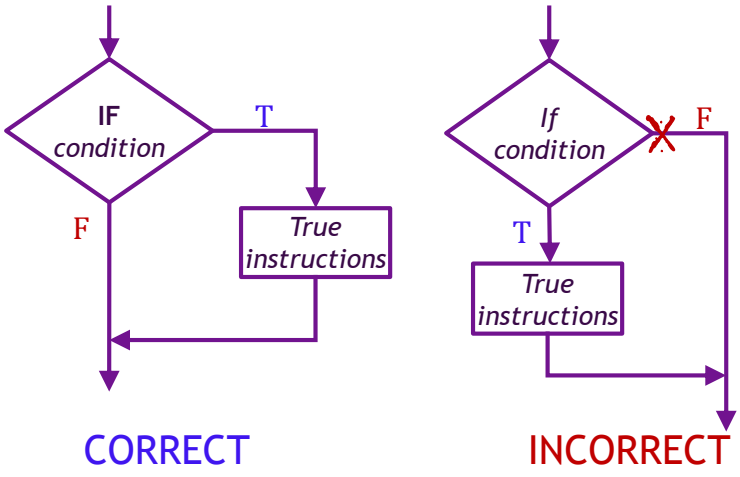




CORRECT

INCORRECT

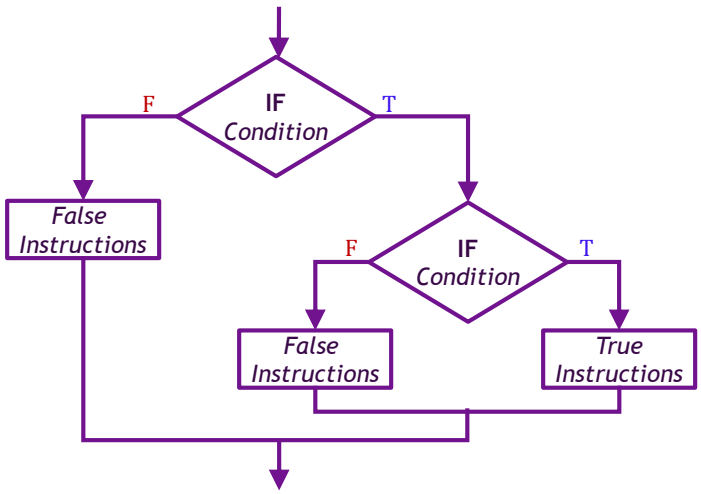
Flowcharting – REVIEW
If-Then Statement



CORRECT

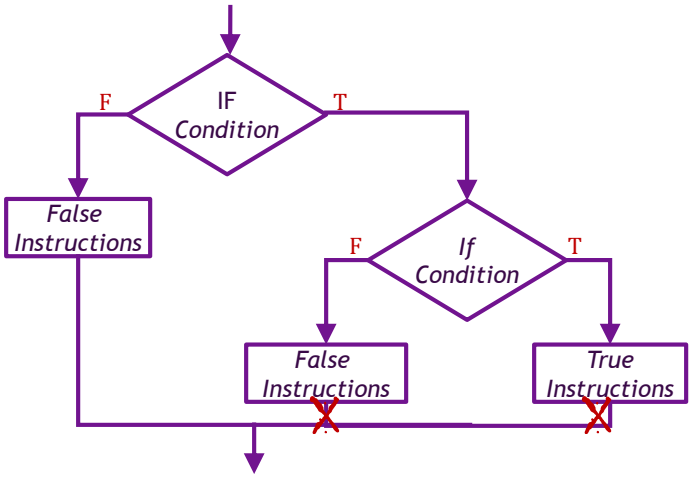
INCORRECT

Nesting If-Then-Else Stmt



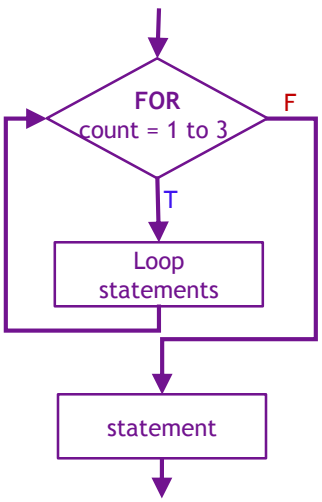
CORRECT

Nesting If-Then-Else Stmt

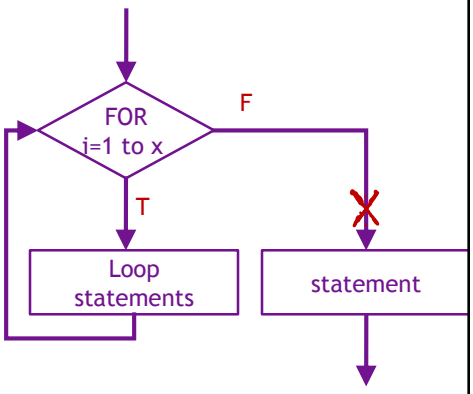


INCORRECT

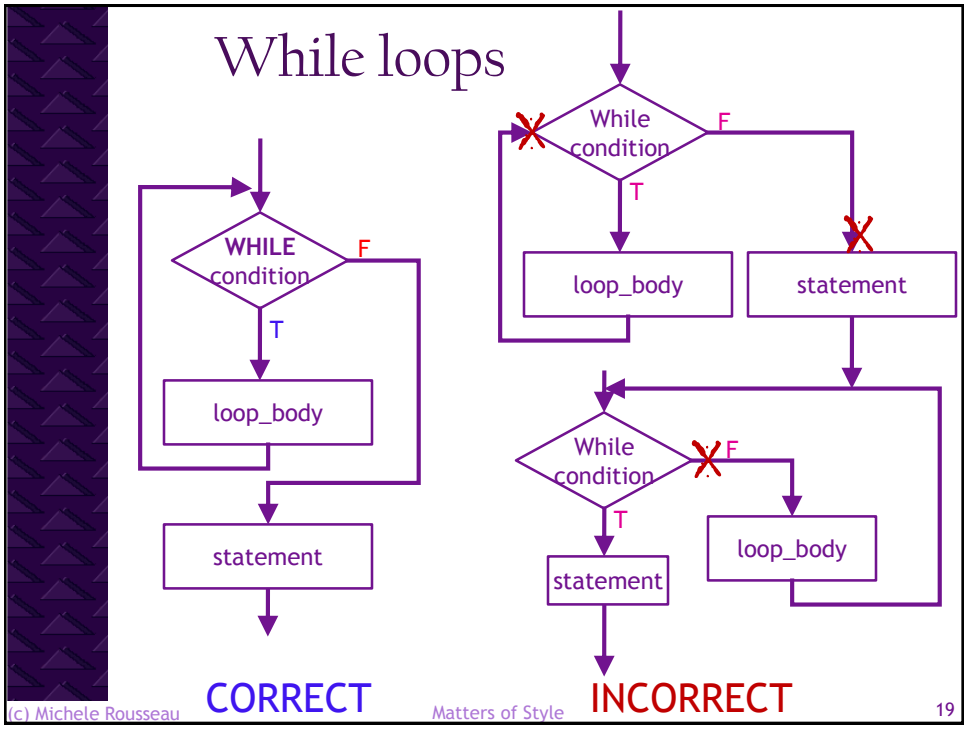
For Loops



CORRECT



INCORRECT


While loops
WHILE condition
F
T
loop_body
statement
CORRECT
While condition
F
T
loop_body
statement
While condition
F
T
loop_body
statement
INCORRECT
(c) Michele Rousseau
Matters of Style
19