# CSC 211: Object Oriented Programming

## Introducing loops (for)

Michael Conti

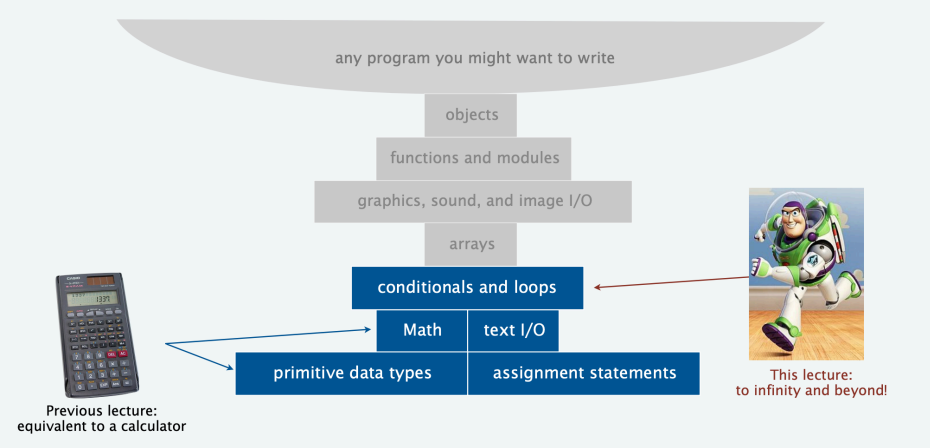
Department of Computer Science and Statistics
University of Rhode Island

Fall 2020

THINK BIG WE DO℠

Original design and development by Dr. Marco Alvarez

# Basic building blocks




https://introcs.cs.princeton.edu/java/lectures/



# Flowchart of `if` statements

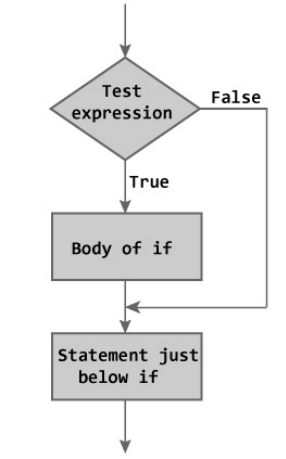



Figure: Flowchart of if Statement

```
// ...
// statements above
// ...

if (test_expression) {
    // body of if
}

// ...
// statements below
// ...
```

https://www.programiz.com/cpp-programming/if-else



# Flowchart of `if` statements

```
// ...
// statements above
// ...

if (test_expression) {
    // body of if
} else {
    // body of else
}

// ...
// statements below
// ...
```

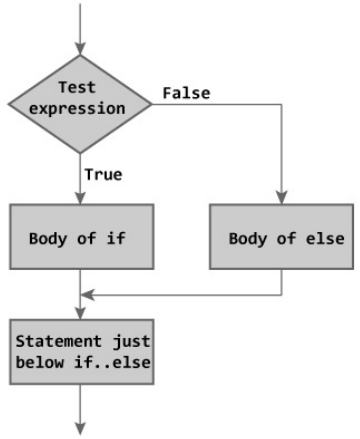
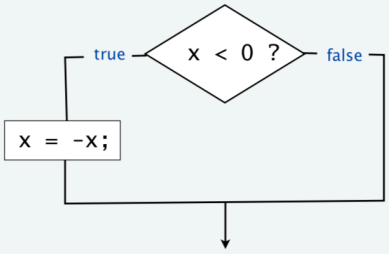



Figure: Flowchart of if...else Statement

https://www.programiz.com/cpp-programming/if-else

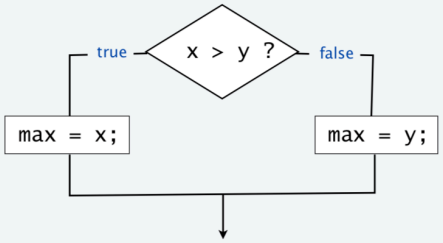
# `if` statement examples

Example: `if (x < 0) x = -x;`




Replaces x with the absolute value of x

Example:
```
if (x > y) max = x;
else       max = y;
```




Computes the maximum of x and y



# The increment/decrement operators

- Increment (++) and decrement (−−) are **unary** operators that add or subtract one, to or from their operand, respectively
  - **pre-increment** and **pre-decrement** operators increment (or decrement) their operand by **1**, and the value of the expression is the resulting incremented (or decremented) value
  - **post-increment** and **post-decrement** operators increase (or decrease) the value of their operand by **1**, but the value of the expression is the operand's original value prior to the increment (or decrement) operation
- Example:

```
int a = 5
5 + ++a
```

```
int a = 5
5 + a++
```



# The increment/decrement operators

```
int a = 5
5 + ++a
```

```
int a = 5
5 + a++
```



# Trace the code

```
int  x;
int  y;

// increment operators
x = 1;
y = ++x;
std::cout << y;
y = x++;
std::cout << y;

// decrement operators
x = 3;
y = x--;
y = --x;
```

# the `for` loop

# Flowchart of `for` statement

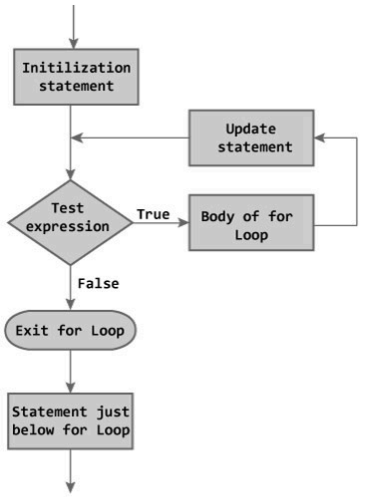



Figure: Flowchart of for Loop

```
// ...
// statements above
// ...

for (init ; test ; update) {
    // body of for
}

// ...
// statements below
// ...
```

https://www.programiz.com/cpp-programming/for-loop



1. initialization

4. update

2. boolean

```
for (int i = 0 ; i < 3 ; i++) {
    std::cout << i << ' ';
}
```

3. statement

then go back to step 2



**A *for* Statement**

```
//Illustrates a for loop.
#include <iostream>
using namespace std;

int main( )
{
    int sum = 0;

    for (int n = 1; n <= 10; n++)  //Note that the variable n is a local
        sum = sum + n;             //variable of the body of the for loop!

    cout << "The sum of the numbers 1 to 10 is "
         << sum << endl;
    return 0;
}
```

*Initializing action*

*Repeat the loop as long as this is true.*

*Done after each loop body iteration*

**Output**

```
The sum of the numbers 1 to 10 is 55
```

from: Problem Solving with C++, 10th Edition, Walter Savitch

# What is the output?

```
int value = 0;

for (int i = 0 ; i < 5 ; i++) {
    value += (i * 10);
}

std::cout << value << std::endl;
```



**for Loop with a Multistatement Body**

**Syntax**

```
for (Initialization_Action; Boolean_Expression; Update_Action)
{
    Statement_1
    Statement_2
       .
       .
       .
    Statement_Last
}
```

Body

**Example**

```
for (int number = 100; number >= 0; number--)
{
    cout << number
         << " bottles of beer on the shelf.\n";
    if (number > 0)
        cout << "Take one down and pass it around.\n";
}
```

from: Problem Solving with C++, 10th Edition, Walter Savitch



# Examples

```
int n = 1;

for ( ; n <= 10 ; n = n + 2)
    std::cout << n << std::endl;

for (n = 10 ; n > 0 ; n -= 2) std::cout << n << std::endl;


for (n = 0 ; n > -30 ; n = n - 7) {
    std::cout << n << std::endl;
}


for (double x = 16.0 ;  x >= 2.0 ;  x = sqrt(x)) {
    std::cout << x;
    std::cout << std::endl;
}
```



# Careful with the semi-colon

- Semi-colon is used to end statements
- Placing it after the parenthesis of a `for` loop creates an **empty statement**

```
for (int count = 1 ; count <= 10 ; count++);
    std::cout << "Hello\n";
```

# pre/post increment/decrement?

```
for (int count = 1 ; count <= 10 ; count++){
    std::cout << count << std::endl;
}
```

V.S

```
for (int count = 1 ; count <= 10 ; ++count){
    std::cout << count << std::endl;
}
```



# Question

- Write a single for loop to print the first 50 even numbers



# Question

- Write a single for loop to print the first 50 even numbers

```
for(int i = 0; i < 100; i++){
    if(i % 2 == 0){
        std::cout << i <<std::endl;
    }

}
```



# Question

- Write a single for loop to print the average of the first 25 multiples of 3

# Question

- Write a single for loop to print the average of the first 25 multiples of 3

```
int multiple = 0;
int sum = 0;

for(int i = 0; i < 25; i++){

    multiple += 3;
    sum += multiple;

}

std::cout << (sum / 25) << std   ::  endl;
```