

For Loops

Sections 5.1, 5.6

for loops (5.6)

- Just like a while loop, but more concise!

```
// initialize      test      update
for (count = 0; count < 5; count++)
{
    cout << count << endl;
}
```

For vs While

This for loop

```
for (count = 1; count <= 9; count += 3)
{
    cout << count << endl;
}
```

is equivalent to what **while** loop?

Example

- Write a `for` loop that outputs odd integers less than 10 and greater than 0

Practice

- What does this output?

```
for (i = 5; i < 10; i += 2)
{
    cout << i;
}
```

- Rewrite the for loop as a while loop

Problem

- Write code that will print the sum of the odd integers between 1 and 50 inclusive.
- Do this with a for loop
- Do this with a while loop

Practice

- Write a program that computes the **factorial** of a number. The factorial of a number is given by the formula below. The user will input a value for N.
- $N! = N * (N-1) * (N-2) * \dots * 2 * 1$
 - Where $0! = 1$, $1! = 1$, $2! = 2$, $3! = 6$, ...

Localized Declarations

```
for (int i = 0; i < n; i++)
```

```
{
```

```
    cout << i << endl;
```

```
}
```

```
cout << i << endl; // This will cause an error
```

- `i` is declared **ONLY** in the loop
- Convert this to a **while** loop

Potential Pitfalls

- What is the output of the following loop

```
for (count = 0; count < 5; count++)  
{  
    cout << count << endl;  
    count++;  
}
```

Problem

- Write a program that allows the user to enter 20 integers, you should then print out the following:
 - The sum of all integers inputted
 - The average of all integers inputted
 - The largest integer of all integers inputted

Increment and Decrement Operators (5.1)

- C++ provides a shortcut to increment or decrement a variable by 1
 - Always by 1

```
int x = 99;
```

```
x++; // this is equivalent to x += 1
```

```
x--; // this is equivalent to x -= 1
```

In a Loop

```
int x = 1;
while( x < 5 ){
    cout << " x : " << x << endl;
    x++; // increment
}
```

For more than just loops

- This can be used in an expression:

`y = x++ + 9;`

What does this mean?

- This can also be used in a conditional

`(x-- > 9)`

What does this mean?

Practice

- Write one statement of code to do each of the following:

```
int x =0 , y=1 ;
```

- Add $x + 9$ to y and increment x by 1
- Add $x * 4$ to y and decrement x by 1

Prefix vs Postfix

- `++x` is *prefix*
 - The `x += 1` happens *before* the expression is evaluated
- `x++` is *postfix*
 - the `x += 1` happens *after* the expression is evaluated

```
int y=0, x=0, z=0;
```

```
x = y++;
```

```
y = ++z;
```

```
z = x ++ + 1;
```

Examples

```
int x = 0, y = 0;
```

```
x = y++ * 2;
```

```
y = ++x / 2;
```

```
x = x++ + 1;
```

```
x = ++x + 1;
```

```
y = (y+ x++) * 2;
```

```
x = y++ + ++x;
```


Practice

- Write a single C++ statement to do each of the following:

```
int y = 0, x = 0, z = 0;
```

- Decrement x by 1 then add $2x$ to y
- Subtract $9x - 1$ from y then decrement x by 1
- Increment x and y each by 1 then add $x+y$ to z