`for` Repetition Structures

Last Time

- We
  - Counter and sentinel-controlled repetitions
  - Type casting
  - Formatting output
  - Top-down, stepwise refinement
- Today we will
  - Examine different ways of writing assignments
  - Learn about the increment and decrement operators
  - Start looking at the `for` repetition structure

Assignment Operators

- C++ provides the ability to abbreviate an assignment operator in which the same variable appears on either side of the operator
- sum = sum + num;
- Can be abbreviated to
- sum += num;

Assignment Operators

- This abbreviation can be done to the following operators
  - `+ - * / %`
- Examples, where `c = 3, e = 4`
  - `c += 7`
  - `e %= 2`
  - `c *= 3`
  - `e /= 4`
  - `e -= 1`

Increment and Decrement Operators

- ++ is the unary increment operator

  x++;

  is the same as x = x + 1;

- -- is the unary decrement operator

  x--;

  is the same as x = x - 1;

Pre-increment vs. post-increment

Pre	Post
k = --x;	k =x--;
k = ++x;	k = x++;
Increment/ decrement x then assign value of x to k	Assign value of x to k, then increment or decrement x

Example

What is the output if i = 2?

```
cout << "Value of x is" << i;
cout << "Value of i++ is" << i++;
cout << "Value of ++i is" << ++i;
cout << "Value of --i is" << --i;
cout << "Value of i-- is" << i--;
```


Operator Precedence

`()`	`L->R`	Parentheses
`++, --, static_cast<type>()`	`L->R`	`Unary`
`++, --, !, +, -`	`R->L`	Negation, Unary
`*,/,%`	`L->R`	Mult, div, mod
`+, -`	`L->R`	Add, Subtract
`<<, >>`	`L->R`	Insertion/extraction
`<, <=, >, >=`	`L->R`	Relational
`==, !=`	`L->R`	Equality
`&&`	`L->R`	And
`||`	`L->R`	Or
`?:`	`R->L`	`Conditional`
`=, +=, -=, *=, /=, %=`	`R->L`	Assignment

For loops

- 3 main things for loops:
  - Initialization of lcv, testing of lcv, updating lcv
- For loops provide a concise way to do this

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


General Format

```
for (initialization expression;
  loop repetition condition;
  update expression)
{
  statements;
}
```


Examples

- Write a for loop that outputs odd numbers less than 10
- Write a program that computes the factorial of a number

Localized Declarations

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

`i` is declared ONLY in the loop

Rewrite using a while loop

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

What does this output?

Problem

- Write a program that will print the sum of the odd integers between 1 and 50 inclusive. Write one program using a while and the other using a for loop.

Problem

- Write a program that allows the user to enter an unknown number of integer values one at a time. When the user enters -999, you are to terminate the loop and print the following:
  - The sum of all integers inputted
  - The average of all integers inputted
  - The largest integer of all integers inputted

Summary

- In today's lecture we covered
  - Abbreviating assignment operators
  - Increment and decrement operators
  - `for` repetition structures
- Readings
  - P. 98 Assignment operators
  - P. 99 - 102 Increment and decrement operators
  - P. 104 - 113 `for` repetition structures