Loops

What loops are and how they are used
How to build various loops
i++, j+=2

i++;

++i;

i--;   

--i;

i+=2;

i-=2;
i++, j+=2

i++; i = i + 1;

++i; i = i + 1;

i--; i = i - 1;

--i; i = i - 1;

i+=2; i+=2;

i-=2;
i++, j+=2

i++;  
i = i + 1;

++i;  
i = i + 1;

i--;  
i = i - 1;

--i;  
i = i - 1;

i+=2;  
i = i + 2;

i-=2;  
i = i - 2;
i++, j+=2

```cpp
cout << i++;
cout << ++i;
cout << i--;  
cout << i--;  
cout << --i;
cout << --i;
cout << (i+=2);
cout << (i-=2);
```
cout << i++;
cout << i;  i = i + 1;

cout << ++i;
i = i + 1;  cout << i;

cout << i--;
cout << i;  i = i - 1;

cout << --i;
i = i - 1;  cout << i;

cout << (i+=2);
i = i + 2;  cout << i;

cout << (i-=2);
i = i - 2;  cout << i;
Loops

What loops are and how they are used
How to build various loops
Computers Excel @

- Doing same thing over and over again
- Software never gets tired
What's a loop?

- Code that may be executed more than once in a row
- Either a fixed number of times
- Or until a specific condition is met
- Each pass through the loop is an *iteration*
While Loop

- Syntax

```
while (condition) {
    statements;
}
```

- If (condition) is true
  - Execute statements inside loop body

- Repeat, until (condition) evaluates to false
While Loop

condition

true

statements

false

EXIT
Examples of:

- Count Controlled Loops
Example 1.1: What prints?

```cpp
int number = 1;

while ( number <= 5 )
{
    cout << "Hello";
    number++;
}

cout << number;
```
Example 1.1: What prints?

```cpp
int number = 1;

while ( number <= 5 )
{
    cout << "Hello";
    number++;
}

cout << number;
```
Example 1.1: What prints?

```cpp
int number = 1; // initialize loop control variable

while ( number <= 5 ) // check loop condition
{
    cout << "Hello";  
    number++; // update loop control variable
}

cout << number;
```
Example 1.1: What prints?

```cpp
int number = 1; // initialization

while ( number <= 5 ) // condition
{
    cout << "Hello";
    number++; // update
}

cout << number;
```
Example 1.1: What prints?

```cpp
int number = 1;

while ( number <= 5 )
{
    cout << "Hello";
    number++;
}

cout << number;
```
Example 1.2: What prints?

```cpp
int number = 6;

while ( number <= 5 )
{
    cout << "Hello";
    number++;  
}

cout << number;
```
int number = 1;

while ( number <= 5 )
{
    cout << "Hello";
}

cout << number;
Example 1.4: What prints?

```c
int number = 1;

while ( number <= 100 )
{
    cout << "Hello";
    number++;
}

cout << number;
```
int number = 1;

while ( number < 100 )
{
    cout << "Hello";
    number++;
}

cout << number;
Example 1.6: What prints?

```cpp
int number = 1;

while ( number <= 100 )
{
    cout << "Hello";
    number++;
}

cout << number;
```
Example 1.7: What prints?

```cpp
int number = 1;

while ( number <= 100 )
{
    cout << "Hello";
    number++;
}

cout << number;
```
Example 2.1: What prints?

```cpp
int x = 1;

while ( x <= 5 )
{
    cout << x << endl;
    x++;  
}

cout << x << endl;
```
Example 2.1: What prints?

```cpp
int x = 1;

while (x <= 5)
{
    cout << x << endl;
    x++;
}
```

```cpp
cout << x << endl;
```
Example 2.1: What prints?

```cpp
int x = 1; // initialize loop control variable

while ( x <= 5 ) // check loop condition
{
    cout << x << endl;
    x++; // update loop control variable
}
```

```cpp
// initialize loop control variable
while ( x <= 5 ) // check loop condition
{
    cout << x << endl;
    x++; // update loop control variable
}
```
Example 2.1: What prints?

```cpp
int x = 1;                // initialization

while ( x <= 5 )           // condition
{
    cout << x << endl;
    x++;                  // update
}

cout << x << endl;        // print the final value of x
```
Example 2.2: What prints?

```cpp
int x = 6;

while ( x <= 5 )
{
    cout << x << endl;
    x++;
}
```

```
 burns 4
  5
  6
  7
```

```
errors 5
```
Example 2.3: What prints?

```cpp
int x = 1;

while ( x <= 5 )
{
    cout << x << endl;
}
```

int x = 1;

while ( x <= 100 )
{
    cout << x << endl;
    x++;    // update loop control variable
}

cout << x << endl;
Example 2.5: What prints?

```cpp
int x = 1;

while ( x < 100 )
{
    cout << x << endl;
    x++;;
}
```

```cpp
cout << x << endl;
```
Example 2.6: What prints?

```cpp
int x = 1;

while ( x <= 100 )
{
    cout << x << endl;
    x++;
}
```

Example 2.7: What prints?

```cpp
int x = 1;

while ( x <= 100 ){
    cout << x << endl;
    x++;
}
```

Example 2.4: What prints?

int x = 1;

while ( x <= 100 )
{
    cout << x << endl;
    x++;
}

cout << x << endl;
Example 2.8: What prints?

```cpp
int x = 100;

while ( x > 0 )
{
    cout << x << endl;
    x--;  
}
```

int x = 100;

while ( x > 0 )
{
    cout << x << endl;
    x--;
}

cout << x << endl;
Example 2.8: What prints?

```cpp
int x = 100;

while ( x > 0 )
{
    cout << x << endl;
    x--;  // update loop control variable
}
```

This code initializes an integer variable `x` to 100 and then enters a `while` loop that continues as long as `x` is greater than 0. Inside the loop, `x` is printed and then decremented by 1. The loop continues until `x` becomes 0, at which point the `while` condition becomes false and the loop terminates.
Example 2.8: What prints?

```cpp
int x = 100;

while ( x > 0 )
{
    cout << x << endl;
    x--;
}
```

int x = 100;

while ( x >= 0 )
{
    cout << x << endl;
    x--; 
}

}
Example 2.8: What prints?

```cpp
int x = 100;

while ( x > 0 )
{
    cout << x << endl;
    x--;  
}
```

Example 2.10: What prints?

```cpp
int x = 100;

while ( x > 0 )
{
    x--;  
    cout << x << endl;
}
```

Example 2.4: What prints?

```cpp
int x = 1;

while ( x <= 100 )
{
    cout << x << endl;
    x++;
}
```
Example 2.11: What prints?

```cpp
int x = 1;

while ( x <= 100 )
{
    if ( x % 13 == 0 )
    {
        cout << x << endl;
    }
    x++;
}
cout << x << endl;
```
Example 3.1: Not count controlled

```cpp
int year;
cin >> year;

while ( year <= 0 )
{
    cout << "Year must be greater than 0"
        << endl
        << "Try again.."
        << endl;

    cin >> year;
}
```

Example 3.1: Not count controlled

```cpp
int year;
cin >> year;

while (year <= 0)
{
    cout << "Year must be greater than 0"
    << endl
    << "Try again.."
    << endl;

    cin >> year;
}
```
int year;
cin >> year; // initialization

while ( year <= 0 ) // check
{
    cout << "Year must be greater than 0"
         << endl
         << "Try again.."
         << endl;

    cin >> year; // update
}
**Example 3.1: Not count controlled**

```cpp
int year;
cin >> year;

while ( year <= 0 )
{
    cout << "Year must be greater than 0"
    << endl
    << "Try again.."
    << endl;

    cin >> year;
}
```
Example 3.2

```c++
int year;
// initialize loop control variable
cin >> year;

while ( cin >> year && year <= 0 )
{
    cout << "Year must be greater than 0"
        << endl
        << "Try again..
        << endl;

    cin >> year;
}
```

Example 4: Average Salary

- Say, don't know how many employees apriori
  - Then loop can't be "count" controlled
- Then can stop loop by entering negative salary
Example 4.1: Average Salary

```cpp
int count = 0;
double total = 0;
double salary;
cin >> salary;

while ( salary >= 0 ) {
    total += salary;
    count++;
    cin >> salary;
}

if ( count > 0 ) {
    cout << "Average is: " << total / count;
} else {
    cout << "No data" << endl;
}
```
int count = 0;
double total = 0;
double salary;
cin >> salary; // initialization

while ( salary >= 0 ) { // check
total += salary;
count++;
cin >> salary; // update
}

if ( count > 0 ) {
cout << "Average is: " << total / count;
} else {
cout << "No data" << endl;
}
Example 4.2: Average Salary

```cpp
int count = 0;
double total = 0;
double salary;
cin >> salary;

while ( cin >> salary && salary >= 0 ) {
    total += salary;
    count++;
    cin >> salary;
}

if ( count > 0 ) {
    cout << "Average is: " << total / count;
} else {
    cout << "No data" << endl;
}
```
Example 5: Average number

- Say, don't know how many numbers apriori
- Also say, numbers can be negative
- Then, can stop loop by entering a non-number
int count = 0;
double total = 0;
double number;

while ( cin >> number ) {
    total += number;
    count++;
}

if ( count > 0 ) {
    cout << "Average is: " << total / count;
} else {
    cout << "No data" << endl;
}
```cpp
int    count = 0;
double total = 0;
double number;

while ( cin >> number ) {
    total += number;
    count++;
}

if ( count > 0 ) {
    cout << "Average is: " << total / count;
} else {
    cout << "No data" << endl;
}
```

alpha will put "cin" into a "fail" state

need to clear before any further input takes place
cin.clear();
Summary

- while loops
- Examples