

Slides #5.2

# While Loops

Chapter 2.4 – 3.4 (some parts)

```
#include <stdio.h>
int main(void)
{
    int count;

    for (count = 1; count <= 500; count++)
        printf("I will not throw paper airplanes in class.");

    return 0;
}
```



How can you write a program to...

- print numbers 1 – 10?
- add prime numbers 1 to 1,000?
- find 15 digit prime numbers?
- simulate 10,000 vaccine ideas?

# Increment and Decrement

- Add 1 to x:

Subtract 1 from x:

- Example:

– int index = 0;

index++; 

– int sum = 100;

sum--; 

Equivalent:

- Prefix vs Postfix:

– b++ and ++b are a little different; we may cover it later. Just don't do silly things:

int plainSilly = ++x \* y-- - x++;

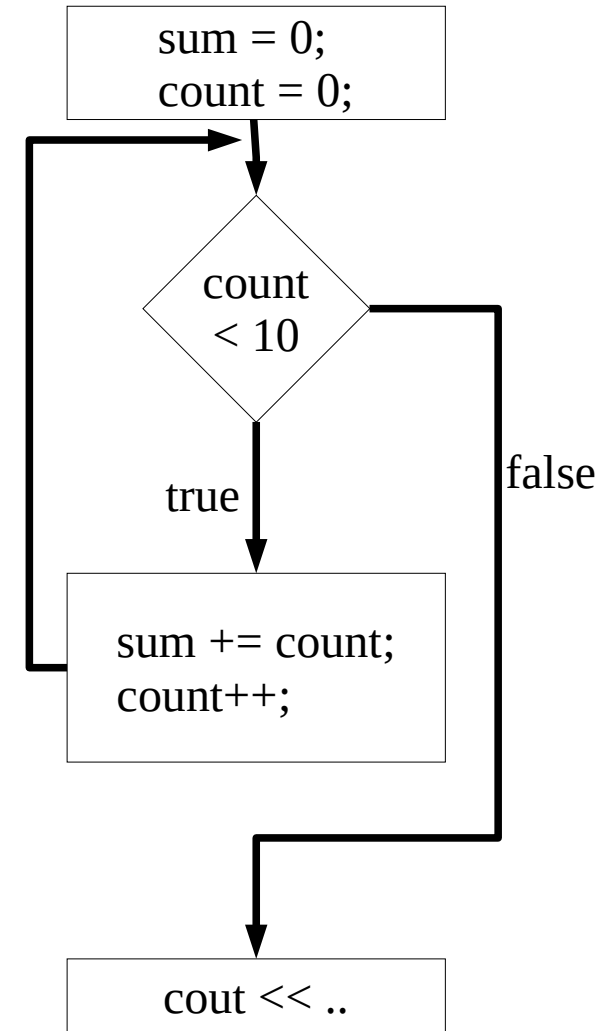
# while

Loop means..

- A while loop executes the body...

```
int sum = 0;  
  
int count = 0;  
while (count < 10) {  
    sum += count;  
    count++;  
}  
  
cout << "Total: " << sum;
```

Condition tested..



# Explain this code

```
#include <iostream>
using namespace std;
```

```
int main()
{
```

```
    const int MAX = 10;
    cout << "Enter value between 1 and 9: ";
    int choice = 0;
    cin >> choice;
```

```
    // ..
```

```
    // ..
```

```
    while (choice >= MAX) {
        cout << "ERROR: Re-enter a value: ";
        cin >> choice;
    }
```

```
        cout << "Great choice of "
            << choice << endl;
```

```
    // ..
```

```
    // ..
```

```
    cout << "#s between 1 and "
        << choice << ":\n";
```

```
    int i = 1;
    while (i <= choice) {
        cout << i << endl;
        i++;
    }
```

```
    return 0;
```

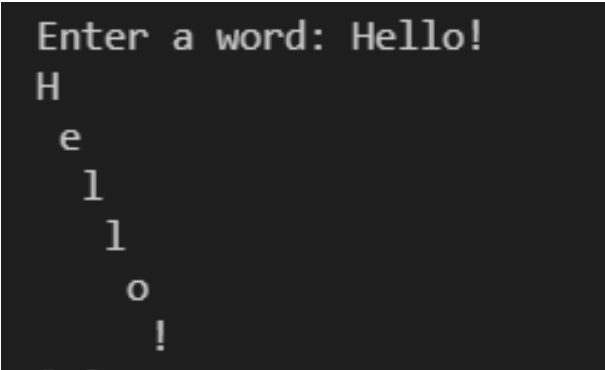
```
}
```

# Looping Through Letters

```
// Print a phrase down the diagonal
#include <iostream>
#include <iomanip>
using namespace std;

int main()
{
    cout << "Enter a word: ";
    string word;
    cin >> word;

    int i = 0;
    while (i < word.length()) {
        cout << setw(i + 1) << word.at(i) << endl;
        i++;
    }
    return 0;
}
```



```
Enter a word: Hello!
H
 e
  l
   l
    o
     !
```

Get the  $i^{\text{th}}$  letter of the word (0 indexed)

# Infinite Loops

- Infinite Loop:..

- Example: a while loop with its condition always true.

```
while (true) {  
    cout << "Still going.....";  
}
```

- or:

```
int index = 0;  
while (index < 10) {  
    cout << "Not done.";  
    index --;  
}
```



# More Infinite Loops

- Mystery Infinite Loop #1:

```
int i = 0;
while (i < 10); {
    cout << "Not done.";
    i ++;
}
```

Loop runs the empty statement an infinite number of times.

- Mystery Infinite Loop #2:

```
int j = 0;
while (j < 10)
    cout << "Not done.";
    j ++;
```

j++ is the first statement after the loop (not part of it).



# Review

- What is printed to the screen?

```
int i = 0;
if (i < 4) {
    cout << i << endl;
} else {
    cout << 4 << endl;
}
```

- What is printed to the screen?

```
int i = 5;
while (i > 0) {
    cout << i;
    i -= 2;
}
```

# Nested Loops

```
// Help space out users in a line
```

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    const int BLANK_SQUARES_BETWEEN_PEOPLE = 6;
```

```
    int nextSquareToStandOn = 1;
```

```
    int choice = 0;
```

```
    while (choice >= 0) {
```

```
        cout << "How many people in your group? (-1 to quit)" << endl;
```

```
        cout << ": ";
```

```
        cin >> choice;
```

```
        int i = 0;
```

```
        while (i < choice) {
```

```
            cout << "Please stand on square " << nextSquareToStandOn << endl;
```

```
            nextSquareToStandOn += BLANK_SQUARES_BETWEEN_PEOPLE;
```

```
            i++;
```

```
        }
```

```
    }
```

```
    return 0;
```

```
}
```

```
How many people in your group? (-1 to quit)
```

```
: 1
```

```
Please stand on square 1
```

```
How many people in your group? (-1 to quit)
```

```
: 3
```

```
Please stand on square 7
```

```
Please stand on square 13
```

```
Please stand on square 19
```

```
How many people in your group? (-1 to quit)
```

```
: -1
```

# Arbitrary Nested Loops

- You can nest loops:

```
int i = 0;
while (i < 3) {
    int j = 0;
    while (j <= i) {
        cout << j;
        j++;
    }
    cout << endl;
    i++;
}
```

What is the output?

# Review

- What will each of these print?

```
const int LOOPS = 5;
int count = 0;
while (count < LOOPS) {
    cout << "Loop " << count << endl;
    count++;
}
```

```
const int LOOPS = 5;
int num = LOOPS;
while (num > 0) {
    int count = num;
    while (count < LOOPS) {
        cout << count << ", ";
        count++;
    }
    cout << endl;
    num --;
}
```

# Suggested Questions

- Write a program which outputs the following:

```
*****
```

```
*****
```

```
*****
```

- What is the output of the following code:

```
int i = 0;
while (i < 4) {
    int j = i;
    while (j >= 0) {
        cout << "$";
        j--;
    }
    cout << "\n";
    i++;
}
```

# Summary

- Loops execute code many times (0 or more)

```
while ( condition ) {  
    myStatements();  
    iWantTo();  
    repeat();  
}
```