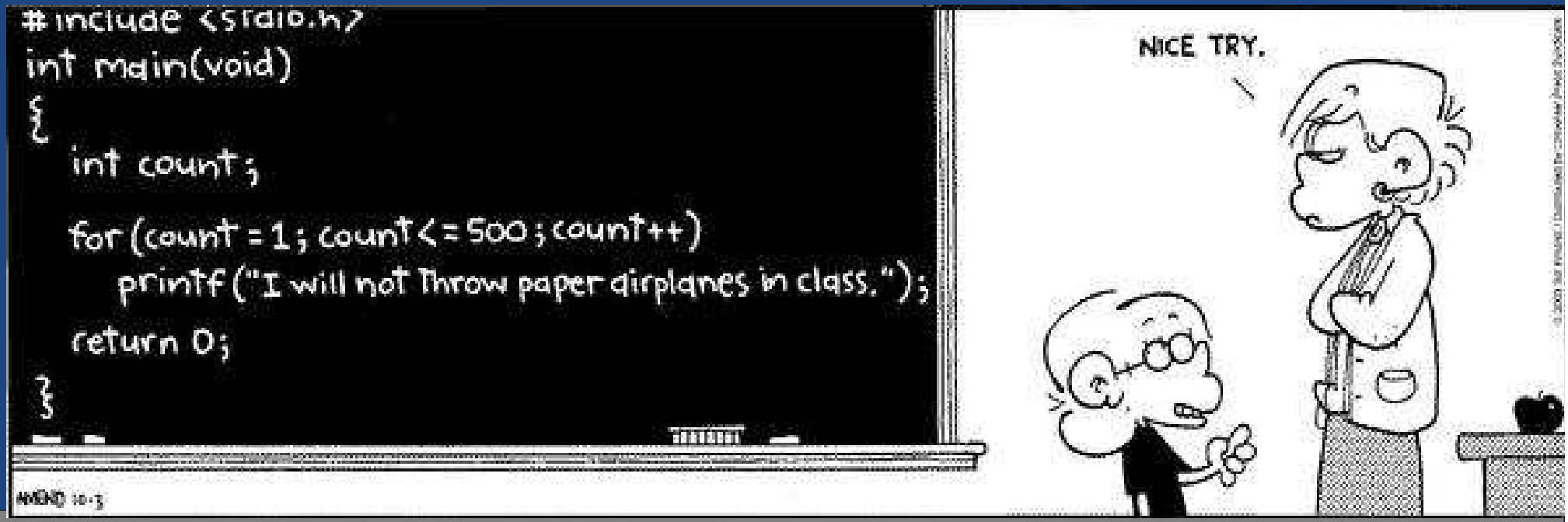
Slides #5.2

While Loops

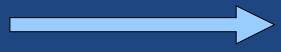
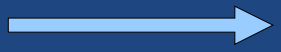
Chapter 2.4 – 3.4 (some parts)



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: Equivalent:
  - int index = 0;
    index++; 
  - int sum = 100;
    sum--; 
- Prefix vs Postfix:
  - x++ and ++x are a little different;
    Just don't do silly things like:
    int plainSilly = ++x * y-- - x++;

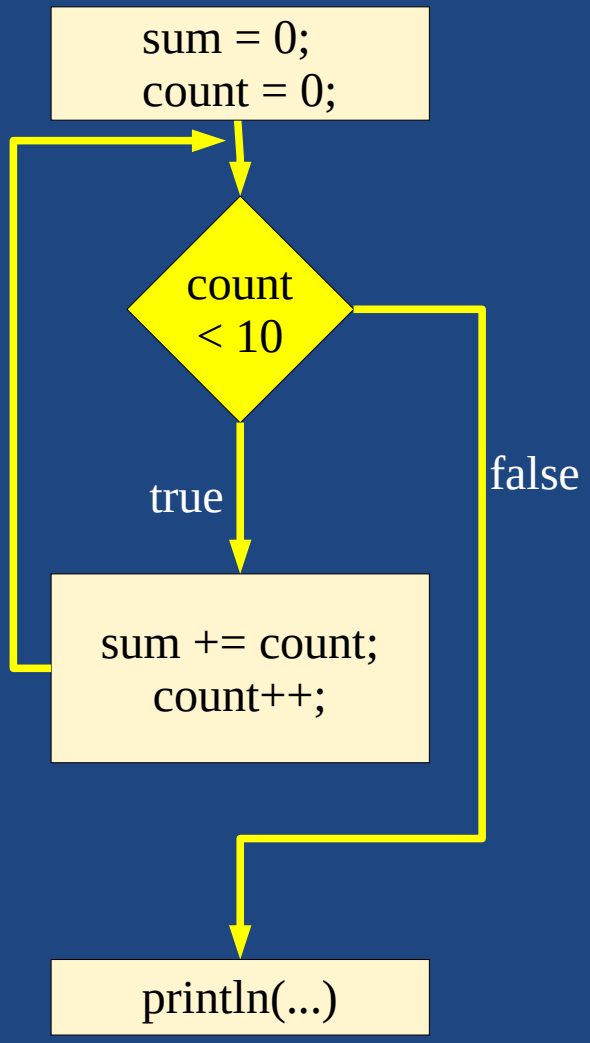
while

Loop means..

- A while loop executes the body...

```
// Add up a sequence of numbers
#include <print>
using namespace std;

int main() {
    // Create the accumulator
    int sum = 0;

    // Loop through values
    int count = 0;
    while (count < 10) {
        sum += count;
        count++;
    }

    // Use the answer
    println("Total: {}", sum);
}
```

While loops test the condition..



Explain this code

```cpp
// Understand the purpose of these loops
#include <iostream>
#include <print>
using namespace std;

int main() {
    const int MAX = 10;

    int choice = 0;
    print("Enter value between 1 and 9: ");
    cin >> choice;

    // ???
    while (choice >= MAX) {
        print("ERROR: Re-enter a value: ");
        cin >> choice;
    }

    println("Great choice of {}", choice);

    // ???
    int i = 1;
    while (i <= choice) {
        println("{}", i);
        i++;
    }
}
```

Looping Through Letters in a string

```cpp
// Print a word vertically
#include <iostream>
#include <print>
#include <string>
using namespace std;

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

    int i = 0;
    while (i < word.length()) {
        println("{}",word.at(i));
        i++;
    }
}
```

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

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

Infinite Loops

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

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

- or:

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

More Infinite Loops

- Mystery Infinite Loop #1:

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

Loop runs the empty statement an infinite number of times.

- Mystery Infinite Loop #2:

```
int j = 0;
while (j < 10)
      println("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) {
    print(“{}”, i);
} else {
    println(“4”);
}
```

- What is printed to the screen?

```
int i = 5;
while (i > 0) {
    print(“{}”, i);
    i -= 2;
}
```

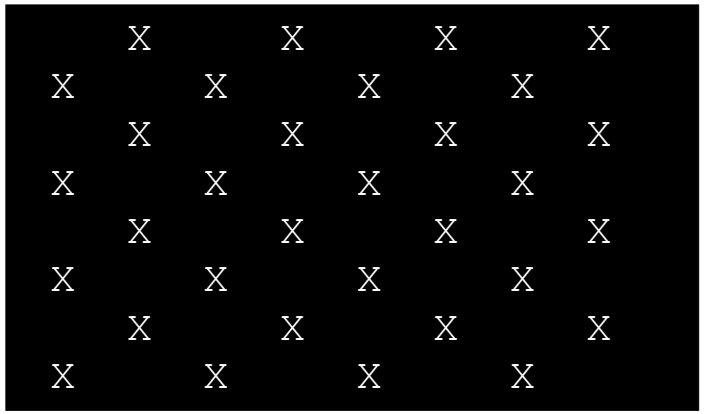
Nested Loops

```
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
```

```
// Space out people standing in a line
#include <iostream>
#include <print>
using namespace std;

const int BLANK_SQUARES = 6;
int main() {
    // Next square for someone to stand on
    int nextSquare = 1;

    int choice = 0;
    while (choice >= 0) {
        println("How many people in your group? (-1 to quit)");
        print(": ");
        cin >> choice;

        int i = 0;
        while (i < choice) {
            println("Please stand on square {:3}", nextSquare);
            nextSquare += BLANK_SQUARES;
            i++;
        }
    }
}
```

Nested Loops

```cpp
// Print a checker board
#include <iostream>
#include <print>
using namespace std;

const int SIZE = 8;

int main() {
    int row = 0;
    while (row < SIZE) {
        int col = 0;
        while (col < SIZE) {
            bool isBlack = (row + col) % 2;
            if (isBlack) {
                print("{:^3}", "X");
            } else {
                print("{:^3}", "");
            }
            col++;
        }
        println();
        row++;
    }
}
```



= checkerBoard.cpp

Arbitrary Nested Loops

- You can nest loops:

```
int i = 0;
while (i < 3) {
    int j = 0;
    while (j <= i) {
        print("{}", j);
        j++;
    }
    println();
    i++;
}
```

What is the output?

Review

- What will each of these print?

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

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

Suggested Questions

- Write a program which outputs the following:

```
*****
*****
*****
```

- What is the output of the following code:

```
int main() {
    int i = 0;
    while (i < 4) {
        int j = i;
        while (j >= 0) {
            print("$");
            j--;
        }
        println();
        i++;
    }
}
```

Summary

- Loops execute code many times (0 or more)

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