#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;
}
Topics

1) How can we work with true and false?
2) How can write if statements?
3) Loops: while and for
Boolean Expressions
Boolean Expressions

• Boolean Expressions evaluate to...
  – Called a “condition.”

• Most often used with if or loop (while, for, do..)

```cpp
int loops = 45;

// Check loop length
if (loops > 30) {
    cout << “This may take a while!\n”;  
}

// Count down
while (loops != 0) {
    cout << “Counting down... “ << loops << endl;
    loops = loops - 1;
}
```
Truth?

#include <iostream>
using namespace std;
int main()
{
    cout << "Dear C++, what is the truth?\n";
    cout << " Is 2 greater than 5? = " << (2 > 5) << endl;
    cout << " Is 2 less than 5? = " << (2 < 5) << endl;
    return (0);
}

• How C++ interprets values:
  – What is false:
    •
  – What is true:
   •

Aside:
You can also display “true” and “false” by first using:
cout << boolalpha;
Equality

• Equality Operators:
  – Equal:
  – Not Equal:

• Examples:
  cout << (0 == 0); // 1
  cout << (65 != 65); // 0
  cout << (42 == 2); // 0
  cout << ('b' != 'a'); // 1

• Equality vs Assignment:
  – if the values are the same (==).
  – the values the same (=).
Relational Operators

• Compare two values to see which is less/greater:
  - `<`  Less than
  - `<=` Less than or equal to
  - `>`  Greater than
  - `>=` Greater than or equal to

• Examples:
  ```cpp
  cout << (age >= 65); // 1 if senior
  if (age >= 65) {...}  // Check for senior
  if (value < 0) {...}  //..
  if (0 > value) {...}  // Also check for neg. value
  if (value <= 0) {...}  // Check for non-positive value
  ```
If
If Statement

• The if statement performs conditional execution once:

    cout << “Value is negative.\n”;  

• Style
  – Indent the “then” statements to make it easy to read.  
    (Compiler does not care!)
  – Good style...
if, if-else

- .. if (age < 18) {
  cout << "Minor";
}

- .. if (age < 18) {
  cout << "Minor";
  } else {
  cout << "Adult";
  }

- .. if (age < 18) {
  cout << "Minor";
  } else if (age < 65) {
  cout << "Adult";
  } else {
  cout << "Senior";
  }

19-09-19  10

! PercentToLetterGrade.cpp
Nested if

if statements can be...

```cpp
int main() {
    cout << "Enter your shoe size: ";
    int shoeSize = 0;
    cin >> shoeSize;

    // Print summary
    if (shoeSize > 0) {
        if (shoeSize > 10) {
            cout << "Big feet!\n";
        } else {
            cout << "Small feet!\n";
        }
    } else {
        cout << "Invalid size!\n";
    }
}
```

// Can code without nested if:
```cpp
// Can code without nested if:
if (shoeSize <= 0) {
    cout << "Invalid\n";
} else if (shoeSize > 10) {
    cout << "Big\n";
} else {
    cout << "Small!\n";
}
```
Multiple statements

- Use {...} to place multiple statements in the if or else:

```cpp
#include <iostream>
using namespace std;
int main()
{
    cout << "Enter your IQ: ";
    int iqValue;
    cin >> iqValue;
    bool genius = false;
    if (iqValue > 150)
        genius = true;
    else
        genius = false;
    cout << "Not so smart\n";
}
```

- This code will...

- Must use block statement {...} with multiple statements.

bool is a data type that stores..
Better code (aside)

- Note there is better way to write the logic:

Previous Solution

```cpp
#include <iostream>
using namespace std;
int main()
{
    cout << "Enter your IQ: ";
    int iqValue;
    cin >> iqValue;
    bool genius = false;
    if (iqValue > 150)
        genius = true;
    else {
        genius = false;
        cout << "Not so smart\n";
    }
}
```

Better Solution

```cpp
#include <iostream>
using namespace std;
int main()
{
    cout << "Enter your IQ: ";
    int iqValue = 0;
    cin >> iqValue;
    bool genius = (iqValue > 150);
    if (!genius) {
        cout << "Not so smart\n";
    }
}
```

! means not

No if required!
Common Errors

- if (a < b); {
  cout << "less\n";
}

- if (a = b) {
  cout << "Equal\n!";
}
Scope

• Variables exist in a scope:
  – Variables defined in a block...

• What are the errors?
  
  ```
  if (a > b) {
      int fromThen = 1;
  } else {
      cout << fromThen;
      int fromElse = 2;
  }
  cout << fromThen;
  cout << fromElse;
  ```
Exercise

- Complete this program to tell the user if their number is negative, positive, or zero.

- 3 Sample outputs:

```cpp
int main()
{
    double favNum = 0;
    cout << "What's the number? ";
    cin >> favNum;

    // Your code here!
}
```

What’s the number? 42
Positive

What’s the number? 0
Zero

What’s the number? -3
Negative
While Loops
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; we may cover it later.
  - For now, don’t do silly things:
    int plainSilly = ++x * y--; - x++;
while

• A while loop executes the body...

```cpp
int sum = 0;
int count = 0;
while (count < 10) {
    sum += count;
    count++;
}
```

```cpp
cout << "Total: " << sum;
```
Explain the following code:

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

int main()
{
    const int MAX = 10;
    cout << "Enter value between 1 and 9: ";
    int inVal = 0;
    cin >> inVal;

    // ..
    // ..
    //..
    while (inVal <= 0 || inVal >= MAX) {
        cout << "ERROR: Re-enter a value: ";
        cin >> inVal;
    }

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

    // ..
    // ..
    cout << "#s between 1 and " << inVal << "::n";
    int i = 1;
    while (i <= inVal) {
        cout << i << endl;
        i++;
    }

    return 0;
}
```

= validateInput.cpp
Infinite Loops

- Infinite Loop:

  Example: a while loop with its condition always true.

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

- or:

  ```cpp
  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?

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

- What is printed to the screen?

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

- You can nest loops:

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

What is the output?
• What will each of these print?

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

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

- Boolean expressions evaluate to true or false.
  - Comparison: ==, !=, >, <, >=, <=

- if-else statement control program branching.
  - Can pick which code to run.

- Loops execute code many times (0 or more)
  ```java
  while ( condition ) {
    myStatements();
    iWantTo();
    repeat();
  }
  ```
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++;
  }