More Complex Versions of the if Statement

Class 13

if-else

- the if-else statement is an expansion of the plain if statement
- as with the if statement, an expression is evaluated to give a Boolean result
- if the Boolean assertion is true, a block of statements is executed
- if the Boolean assertion is false, however, the two statements differ

if-else

- the if-else statement is an expansion of the plain if statement
- as with the if statement, an expression is evaluated to give a Boolean result
- if the Boolean assertion is true, a block of statements is executed
- if the Boolean assertion is false, however, the two statements differ

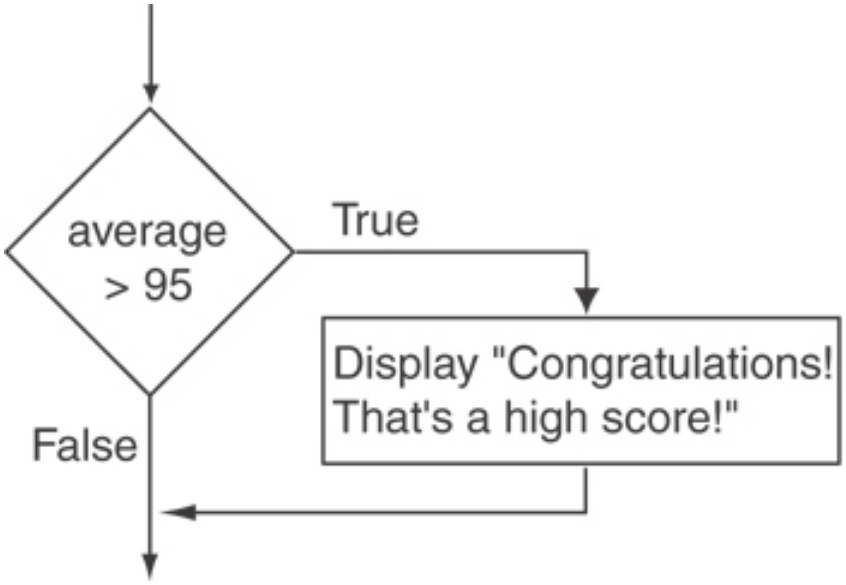
- if the Boolean assertion is false:
  - in the simple if statement, the if statement terminates
  - in the if-else statement, an alternate block of statements (the else block) executes instead of the if block
- the simple if asks a yes-no question
- the if-else asks a this-or-that question

if-else

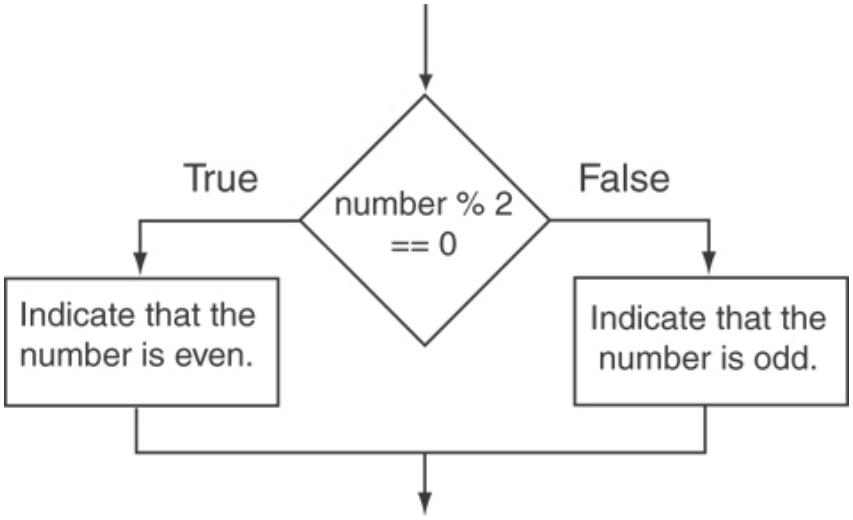
```
cout << "Enter an integer: ";
int number;
cin >> number;

if (number % 2 == 0)
{
  cout << number << " is even" << endl;
}
else
{
  cout << number << " is odd" << endl;
}
```

- the plain if statement is like a detour in the road
- the if-else statement is like a fork in the road



simple if flowchart



if-else flowchart

if-else

```
if (number % 2 == 0)
{
  cout << number << " is even" << endl;
}
else
{
  cout << number << " is odd" << endl;
}
```

- if-else determines whether this (number is even)
- or that (number is odd)
- how many questions do you have to ask to determine whether a number is even or odd?

if-else

```
if (number % 2 == 0)
{
  cout << number << " is even" << endl;
}
else
{
  cout << number << " is odd" << endl;
}
```

- if-else determines whether this (number is even)
- or that (number is odd)
- how many questions do you have to ask to determine whether a number is even or odd? ONLY ONE!

Scope

- the body of an if statement is a new scope
- a variable declared within an if body is not visible outside that scope
- here is a common mistake:

```
if (foo)
{
  int bar = 10;
}
else
{
  int bar = 20;
}

cout << bar << endl; // error, bar is not in scope
```

Scope

- instead, there are two solutions
- slight pros and cons of each

```
int bar;

if (foo)
{
  bar = 10;
}
else
{
  bar = 20;
}
// ok, bar is still in scope
cout << bar << endl;
```

```
if (foo)
{
  int bar = 10;
  cout << bar << endl;
}
else
{
  // ok, this is a different bar
  int bar = 20;
  cout << bar << endl;
}
```

if Style

- the style guide is explicit about the formatting of if statements
  1. the "i" of if, and the "{", and "}" of the if block of a simple if statement shall be vertically aligned in the same column
  2. the "i" of if, the "e" of else, and both "{"s and "}"s of the if and else blocks of an if-else statement shall be vertically aligned
  3. the statement(s) in the if block, and the else block if present, shall be indented two spaces from the "i" of the if
  4. every if and else block shall have curly braces, even if it consists of a single statement

Style Example

```
cout << "Enter the dividend and the divisor ";
int dividend;
int divisor;
cin >> dividend >> divisor;

if (divisor == 0)
{
  cout << "Division by zero is not defined" << endl;
  cout << "Please try again" << endl;
}
else
{
  int quotient = dividend / divisor;
  cout << dividend << " divided by " << divisor
    << " is " << quotient << endl;
}
```

if-else if Statement

- the simple if is for zero or one alternative paths
- the if-else is for two alternative paths
- sometimes there are more than two alternative paths

- C++ allows for as many alternative paths as are needed
- the alternatives must be non-overlapping
- examples
  - your letter grade based on what your score is
  - your speeding fine based on how fast you were driving
  - the hurricane category based on how hard the wind is blowing

```
if (score >= A_CUTOFF)
{
  grade = 'A';
}
else if (score >= B_CUTOFF)
{
  grade = 'B';
}
else if (score >= C_CUTOFF)
{
  grade = 'C';
}
else if (score >= D_CUTOFF)
{
  grade = 'D';
}
else
{
  grade = 'F';
}
```

Number of Questions

- in the previous slide, how many categories are there?

Number of Questions

- in the previous slide, how many categories are there?
  5
- how many Boolean questions were needed?

Number of Questions

- in the previous slide, how many categories are there?
  5
- how many Boolean questions were needed?
  4

- in general one fewer questions are needed than categories
- the following is wrong:

```
else if (score >= F_CUTOFF)
{
  grade = 'F';
}
```

Nested ifs

- some situations are more complicated
- sometimes, the situations overlap
- you use if-else if statements for non-overlapping or mutually exclusive alternatives

- at a bank, everyone qualifies for the standard loan rate
- if you are employed, you qualify for a special rate
- if you are employed and you have graduated from college, you qualify for the super-special rate

Nested ifs

- some situations are more complicated
- sometimes, the situations overlap
- you use if-else if statements for non-overlapping or mutually exclusive alternatives

- at a bank, everyone qualifies for the standard loan rate
- if you are employed, you qualify for a special rate
- if you are employed and you have graduated from college, you qualify for the super-special rate

- to sort this out requires more than an if-else or if-else if
- one way is to employ a nested if structure

Nested ifs

```
if (employed == 'Y')
{
  if (graduated == 'Y')
  {
    cout << "You qualify for the super-special rate!!" << endl;
  }
  else
  {
    cout << "You qualify for the special rate!" << endl;
  }
}
else
{
  cout << "You qualify for our regular rate." << endl;
}
```

Censored

- do not look at Gaddis from the middle of page 175 to the end of page 177
- it will hurt your brain
- it's the wrong way to do this
- it might lead you into bad habits
- it's hard to un-see something, so best not to look in the first place