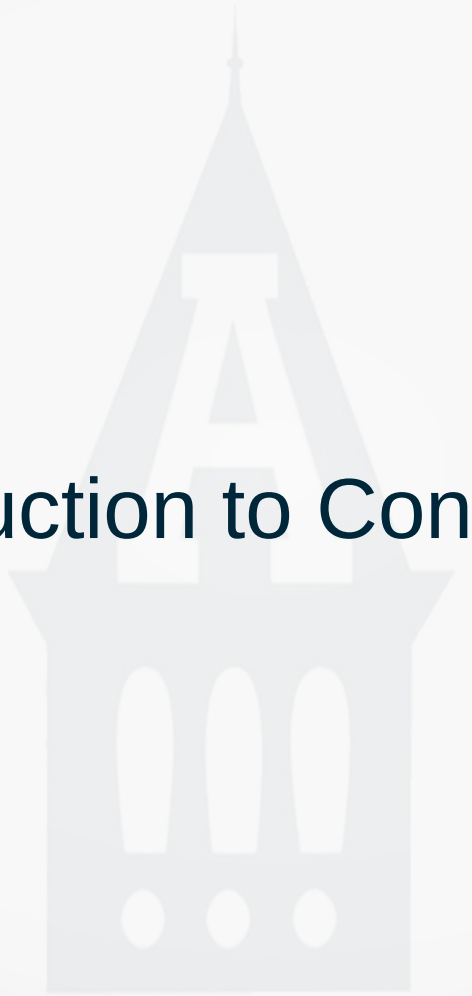
CS 3460



Introduction to Conditionals

Conditionals

- Almost the same as Java at base level, but with some interesting additional capabilities
- Types
  - `if`
  - `if else`
  - `switch`
  - Ternary operator `? :`

Conditionals

- Java requires if conditions evaluate to a boolean
  - `true`/`false`
- C++ allows either a boolean or numeric value
  - If numeric, non-zero is `true`, otherwise `false`

```
if (0)      // evaluates to false
if (0.00)   // evaluates to false
if (1)      // evaluates to true
if (2)      // evaluates to true
if (4.44)   // evaluations to true
```

Conditionals – Initializers

- Both `if` and `switch` allow for an initialization statement

```
if (std::string message = getMessage(); message.size() > 0)
{
    std::cout << message << std::endl;
}
```

- First part is the initialization statement
  - `message` has scope only for the `if` statement
- Second part, right of the `;` is the conditional statement

Conditionals – Switch Statements

- A few differences from Java
  - Can not switch on strings, result must be integral or an enumeration
  - Allows for an initializer
  - `[[fallthrough]]` attribute

```
switch (int input = getUserInput(); input)
{
    case 1:
        std::cout << "1 selected" << std::endl;
        break;
    case 2:
        std::cout << "2 selected" << std::endl;
        break;
    default:
        std::cout << "something else selected" << std::endl;
}
```

Conditionals – Comparing Strings

- Comparing strings in C++ is different from Java
  - In Java the `==` operator compares references, instead use the `.equals` or `.compareTo` methods
  - In C++ the `==` operator compares the contents
    - Why? We'll learn the details soon enough!

```
std::string message1 = "Hello World!";
std::string message2 = "Hello ";
message2 += "World!";

if (message1 == message2) // Resolves to true
```

Comparing Strings – Code Demo