Chapter 4
Making Decisions

- Reading: Chapter 4 (4.1 pp. 149-154; 4.2 pp. 154-162; 4.3 pp. 162-165)
Conditionals

- So far, we can Input, Output and Calculate
- How can we explore relationships between data?
- How can our program only do certain things sometimes?

Decisions

- Relational Expressions allow our program to make a decision
  - based on the data in the program
- What are some decisions we might want our program to make?
Relational Operators

<table>
<thead>
<tr>
<th>Operator</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;</td>
<td>Greater than</td>
</tr>
<tr>
<td>&lt;</td>
<td>Less than</td>
</tr>
<tr>
<td>&gt;=</td>
<td>Greater than or equal to</td>
</tr>
<tr>
<td>&lt;=</td>
<td>Less than or equal to</td>
</tr>
<tr>
<td>==</td>
<td>Equal to</td>
</tr>
<tr>
<td>!=</td>
<td>Not equal to</td>
</tr>
</tbody>
</table>

- All operators are binary
- Each operator is left associative. What does this mean?

Operator Precedence (highest to lowest)

<table>
<thead>
<tr>
<th>Unary plus &amp; minus</th>
<th>+ -</th>
<th>Left associative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiplication, division, and modulus</td>
<td>* / %</td>
<td>Left associative</td>
</tr>
<tr>
<td>Addition &amp; subtraction</td>
<td>+ -</td>
<td>Left associative</td>
</tr>
<tr>
<td>Relational operators</td>
<td>&lt; &lt;= &gt; &gt;=</td>
<td>Left associative</td>
</tr>
<tr>
<td>Relational operators</td>
<td>== !=</td>
<td>Left associative</td>
</tr>
<tr>
<td>Assignment</td>
<td>=</td>
<td>Right associative</td>
</tr>
</tbody>
</table>
The if Statement

- We execute each statement in our program in order
- What if we only want to execute a statement sometimes?
- The if statement
- Single-alternative if

```cpp
if (condition)
{
    // statements
}
```

Practice

```cpp
int age;
const int VOTING_AGE = 18;

cout << "Enter your age:"
    cin >> age;

if (age >= VOTING_AGE)
{
    cout << age << " >= " << VOTING_AGE;
    cout << " You can vote!" << endl;
}
```

What values of age need to be entered to fully test this code?
Write a C++ program segment that allows the user the ability to enter 3 exam scores one at a time. If the average exam score is higher than 60, output “You Passed”; otherwise output “You Failed”. Sum the scores as they are entered.

1. What variables (including their types) are needed in this program segment?

   Hint: If the variable count contains the value 0, what is the value of count after executing the statement count = count + 1;

2. What do you need to do conditionally?

3. What data is necessary to use in the condition?

Problem

Your local bookstore has asked you to write a program to help them determine the cost of shipping of customer orders.

1. If the order is $30 or less then shipping will cost $5

2. If the order is over $30 then shipping will be $3.

Write a C++ program to solve this problem
Problem Modification

The bookstore has now changed its shipping policy so that

1. If the order is $30 or less, shipping is $5
2. If the order is over $30 but less than $50, shipping is $3
3. If the order is $50 or more then shipping is $2

What would we need to change in the program?