C++ If Construct

Instructor: Chi Tse (Ricky)
Computer can make decisions, such as:

- Display a letter grade based on the numerical value
- Display greeting Mr. or Ms. based on user’s gender
- Display AM or PM based on time
- Display advertising based on interest (google/facebook)
- Check if user entered a matching username and password
- (and much more…)

So how do we get our programs to decide things?
If something has happened, we can make computer to perform a task.

Example:
- If number is odd, print out that number is odd.
- If grade is greater than 93, the student receives an A in class.
- If password doesn’t match, ask the user to enter it again.
• Model:
  if ( comparison_is_true )
  {
    //do something here
  }

Example:
if ( number % 2 == 1 )
{
  cout << number << " is odd." << endl;
}
## Conditional Comparison

<table>
<thead>
<tr>
<th>What I want to compare</th>
<th>Code in C++</th>
</tr>
</thead>
<tbody>
<tr>
<td>Is x equal to y?</td>
<td>x == y</td>
</tr>
<tr>
<td>Is x not equal to y?</td>
<td>x != y</td>
</tr>
<tr>
<td>Is x less than y?</td>
<td>x &lt; y</td>
</tr>
<tr>
<td>Is x less than or equal to y?</td>
<td>x &lt;= y</td>
</tr>
<tr>
<td>Is x greater than y?</td>
<td>x &gt; y</td>
</tr>
<tr>
<td>Is x greater than or equal to y?</td>
<td>x &gt;= y</td>
</tr>
</tbody>
</table>
Single statement vs. Block of Code

- There is a shortcut build into if statements:
  - If you only have one instruction to execute
    
    ( ending with ;)
  - You do not need to type out the {} enclosing curly braces

- Example:
  
  if ( number % 2 == 1 )
    cout << number << " is odd." << endl;
If statement using a block of code

- Sometimes multiple instructions need to be executed if a condition is true, in which case the { } enclosing block MUST be provided.
- Suggestion: know both ways, and always use { }
- Example:
  ```
  if ( number % 2 == 1 )
  {
    cout << number;       //first
    cout << " is odd." << endl; //second
  }
  ```
If our initial query was not true, what else can we do?

- We simply do nothing
- or
- We can fall back to a default – else
- or
- We ask another question – else if
If / Else Model

- Model:
  ```
  if ( comparison_is_true )
  {
      //do something here
  }
  else
  {
      //do something else here
  }
  ```
Example:

```cpp
if ( number % 2 == 1 )
{
    cout << number << " is odd." << endl;
}
else
{
    cout << number << " is even." << endl;
}
```
If / Else If / Else Model

Model:

```java
if ( comparison_is_true )
{
    //do something here
}
else if ( another_comparation_is_true )
{
    //do something else here
}
else
{
    //do yet something else here
}
```

Note: else if block can be repeated as many times as needed to detect all the conditions that needs to be picked up.
Example:
if ( number % 3 == 0 )
{
    cout << number << " is divisible by 3." << endl;
}
else if ( number % 3 == 1 )
{
    cout << number << " has remainder of 1." << endl;
}
else
{
    cout << number << " has remainder of 2." << endl;
}
Chain of if / else if statements

- It seems like if and else if serve the same purpose
- Example 1, using if only:
  ```cpp
  if ( number % 3 == 0 )
      cout << "divisible by 3" << endl;
  else if ( number % 3 == 1 )
      cout << "remainder of 1" << endl;
  else if ( number % 3 == 2 )
      cout << "remainder of 2" << endl;
  ```
- What would be the output if number was 5?
Are if statements enough?

• Example 2, using only if statements:
  
  if ( grade >= 97 )
    cout << "A+" << endl;
  
  if ( grade >= 92 )
    cout << 'A' << endl;
  
  if ( grade >=90 )
    cout << "A-" << endl;

• What’s this output if the grade is 100?
When chaining is required

• Example 2, using if and else if statements:
  if ( grade >= 97 )
      cout << "A+" << endl;
  else if ( grade >= 92 )
      cout << 'A' << endl;
  else if ( grade >= 90 )
      cout << "A-" << endl;

• What’s this output if the grade is 100?
Another chaining example

Multiple if statements may be chained by using if and else-if statements as follows:

```cpp
if (x % 5 == 0)
    cout << "x is divisible by 5." << endl;
else if (x % 3 == 0)
    cout << "x is divisible by 3." << endl;
else if (x % 2 == 0)
    cout << "x is divisible by 2." << endl;
else
    cout << "x is not divisible by 2, 3 or 5." << endl;
```
Nesting Statements

- All statements can be used as part of if-else statements.
- Single if-else statement can also be nested another if or else if statement.
  ```java
  if (boolean_condition) {
    if (boolean_condition)
      do something
    else
      do something else
  }
  ```
- Else if … else are also nested in the similar manner.