Life is Full of Alternatives
Part 2

Last Time

• We covered
  o UML activity diagrams
  o Simple if selection structure
  o Relational and equality operators
  o Logical operators
• Today we will look at the if selection structure in more detail

Let's Review

• 07.1 Write the C++ expressions for each of the following
  o x and y are greater than z
  o x is equal to 1.0 or 3.0
  o x is the range z to y inclusive
  o x is outside the range z to y

Single Alternative if

• The if selection structures we saw last time all have a single alternative
  if (condition) or if (condition)
  one statement;
  next statement;
  multiple statements;
  next statement;
• If condition is true, statement(s) following if execute
• If condition is false, statement(s) following if are skipped.

Multiple Statements

• If you have multiple statements that need to be executed if the condition is true, they should be surrounded by curly braces { }

Examples

if (x >= 0)
    cout << "x is positive" << endl;

if (x < 0)
    x = -x;

if ((x == 0) && (y == 0))
{
    x = 1;
    y = 1;
}
Program

• 07.2 Write a program segment that allows the user to input two integer values into variables num1 and num2. Your program is to then exchange the values in the variables num1 and num2 only if num1 is greater than num2.

if/else Selection Structure

• This is the multiple alternative if
• Used when different statements should execute if the condition is false

```
if (condition)
    statementT;
else
    statementF;
```

Examples

```
x = 25.0;
if (y != (x - 10.0))
    x = x - 10.0;
else
    x = x / 2.0;
if ((y < 15.0) && (y >= 0.0))
    x = 5 * y;
else
    x = 2 * y;
```

Program

• 07.3 Write a program that inputs an integer number and outputs if its even or odd
• 07.4 Write a program that computes the area of a triangle or a rectangle based on the user typing in ‘t’ or ‘r’ first

Problem

• 07.5 What is the output of the following program segment

```
i = 5;
j = 2;
if((i % j) == 0)
i = j;
j = i;
cout << i << j;
cout << "That's all folks" << endl;
```
Conditional Operator ?:

• C++ provides the conditional operator ?: as a shortcut way of writing simple if/else structures

• For example, the structure
  ```cpp
  if (x >= 0)
    cout << "positive" << endl;
  else
    cout << "negative" << endl;
  ```

• Could be written as
  ```cpp
  cout << ( x >= 0 ? "positive" : "negative" ) << endl;
  ```

Conditional Operator ?:

• The format of the operator is
  ```cpp
  ( condition ? true-statement : false-statement )
  ```

• The conditional operator works if there is only one statement for a true evaluation and only one statement for a false evaluation

Nested if/else Selection Structures

• What if there are more than two alternatives?
  ```cpp
  if (condition1)
    statement1;
  else if (condition2)
    statement2;
  ...
  else
    default statement;
  ```

Problem

• 07.6 Write a C++ program segment that allows the user the ability to input an integer from the keyboard. If the integer is positive, increment a variable `poscount` by 1. If the integer is negative, increment a variable `negcount` by 1. If neither, increment `zerocount` by 1

Solution

```cpp
cin >> intvalue;
if(intvalue > 0)
  poscount = poscount + 1;
else if(intvalue < 0)
  negcount = negcount + 1;
else
  zerocount = zerocount + 1;
```

• 07.7 Can you come up with another way of doing this?

Solution

```cpp
07.8 Will this solution work?
```

```cpp
cin >> intvalue;
if(intvalue > 0)
  poscount = poscount + 1;
if(intvalue < 0)
  negcount = negcount + 1;
if(intvalue = 0)
  zerocount = zerocount + 1;
```
Problem

• 07.9 Write a program that displays a letter grade corresponding to an exam score
  
  90 - 100 A
  80 - 89 B
  70 - 79 C
  60 - 69 D
  0-59 F

Summary

• In today’s lecture we covered
  
  o if/else selection structures
  o if structures with multiple statements {}
  o Nested if/else selection structures

• Readings
  
  o P. 77 - 78: if/else selection structures
  o P. 78 - 79: conditional operator ?:
  o P. 79 - 81: nested if/else selection structures and if structures with multiple statements