
Life is Full of Alternatives Part 2

09/15/04

CS150 Introduction to Computer Science 1

1

Last Time

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

09/15/04

CS150 Introduction to Computer Science 1

2

Evaluating Expressions: And &&

- `(expr1) && (expr2)`
- For the complete expression to be true, both `expr1` and `expr2` have to be true
- Example:
 - `(temp > 90.0) && (humidity > 0.9)`
 - These are unbearable heat and humidity conditions
 - Both must be true for the entire expression to be true

09/15/04

CS150 Introduction to Computer Science 1

3

Evaluating Expressions: Or ||

- `(expr1 || expr2)`
- The complete expression is true if either `expr1` or `expr2` is true
- Examples:
 - `(salary < minSalary) || (dependents > 5)`
 - To qualify for financial aid, salary has to be less than some minimum salary or the number of dependents is greater than 5
 - Only one condition has to be true

09/15/04

CS150 Introduction to Computer Science 1

4

Evaluating Expressions: Not !

- `!expr`
- Unary operator
- Examples:
 - `!((salary < minSalary) && (dependents > 5))`
 - What makes this true? False?

09/15/04

CS150 Introduction to Computer Science 1

5

Operator Precedence

- We have now added relational, equality and logical operators to the mathematical operators that were introduced last week
- Where do the new operators fit in the precedence table?

09/15/04

CS150 Introduction to Computer Science 1

6

Operator Precedence & Associativity

• ()	L->R	Parentheses
• !, +, -	R->L	Negation, Unary +, -
• *, /, %	L->R	Mult, div, mod
• +, -	L->R	Add, Subtract
• <<, >>	L->R	Insertion/extraction
• <, <=, >, >=	L->R	Relational
• ==, !=	L->R	Equality
• &&	L->R	And
•	L->R	Or
• =	R->L	Assignment

09/15/04

CS150 Introduction to Computer Science 1

7

Expression Evaluation

- According to the operator precedence and associativity rules given on the previous slide, how will the following expressions be evaluated?
 - `x < min + max`
 - `min <= x && x <= max`
 - `!x == y + 2`
 - `x = a + b % 7 * 2`

09/15/04

CS150 Introduction to Computer Science 1

8

bool Data Type

- `bool`: boolean
- Variables of type `bool` can be either `true` or `false`
 - They cannot be any other value
- Boolean variable names should start with `b`
 - See coding standards
- Example

```
bool bCanVote;
int age;
cin >> age;
bCanVote = age >= 18;
cout << bCanVote;
```

09/15/04

CS150 Introduction to Computer Science 1

9

Data Types

- So far we have been introduced to the following C++ data types
 - `int`
 - `double`
 - `char`
 - `string`
 - `bool`

09/15/04

CS150 Introduction to Computer Science 1

10

Examples

- Assume that
 - `double x = 3.0;`
 - `double y = 4.0;`
 - `double z = 2.0;`
 - `bool bFlag = false;`
- What is the value of the following expressions

```
!bFlag
x + y/z <= 3.5
!bFlag || (y + z >= x - z)
!(bFlag || (y + z >= x - z))
```

09/15/04

CS150 Introduction to Computer Science 1

11

Let's Review

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

09/15/04

CS150 Introduction to Computer Science 1

12

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.

09/15/04

CS150 Introduction to Computer Science 1

13

Multiple Statements

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

09/15/04

CS150 Introduction to Computer Science 1

14

Examples

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

if (x < 0)
  x = -x;

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

09/15/04

CS150 Introduction to Computer Science 1

15

Program

- 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

09/15/04

CS150 Introduction to Computer Science 1

16

if/else Selection Structure

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

```
if (condition)      if (condition)
  statementT;      {
else                statementsT;
  statementF;      }
                  else
                  {
                  statementsF;
                  }
```

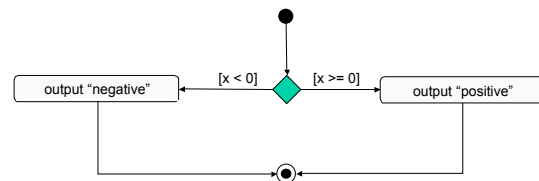
09/15/04

CS150 Introduction to Computer Science 1

17

UML Activity Diagram

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



09/15/04

CS150 Introduction to Computer Science 1

18

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;
```

09/15/04

CS150 Introduction to Computer Science 1

19

Program

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

09/15/04

CS150 Introduction to Computer Science 1

20

Problem

- 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;
```

09/15/04

CS150 Introduction to Computer Science 1

21

Conditional Operator ?:

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

- For example, the structure

```
if (x >= 0)
    cout << "positive" << endl;
else
    cout << "negative" << endl;
```

- Could be written as

```
cout << ( x >= 0 ? "positive" : "negative" ) << endl;
```

09/15/04

CS150 Introduction to Computer Science 1

22

Conditional Operator ?:

- The format of the operator is
 - (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

09/15/04

CS150 Introduction to Computer Science 1

23

Nested if/else Selection Structures

- What if there are more than two alternatives?

```
if (condition1)
    statement1;
else if (condition2)
    statement2;
...
else
    default statement;
```

09/15/04

CS150 Introduction to Computer Science 1

24

Problem

- 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

09/15/04

CS150 Introduction to Computer Science 1

25

Solution

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

- Can you come up with another way of doing this?

09/15/04

CS150 Introduction to Computer Science 1

26

Solution

- Will this solution work?

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

09/15/04

CS150 Introduction to Computer Science 1

27

Problem

- 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

09/15/04

CS150 Introduction to Computer Science 1

28

Summary

- In today's lecture we covered
 - `if/else` selection structures
 - `if` structures with multiple statements `{ }`
 - Nested `if/else` selection structures
- Readings
 - P. 77 - 78: `if/else` selection structures
 - P. 78 - 79: conditional operator `?:`
 - P. 79 - 81: nested `if/else` selection structures and `if` structures with multiple statements

09/15/04

CS150 Introduction to Computer Science 1

29