CSC 211: Object Oriented Programming

Expressions and Selection Statements

Marco Alvarez

Department of Computer Science and Statistics
University of Rhode Island

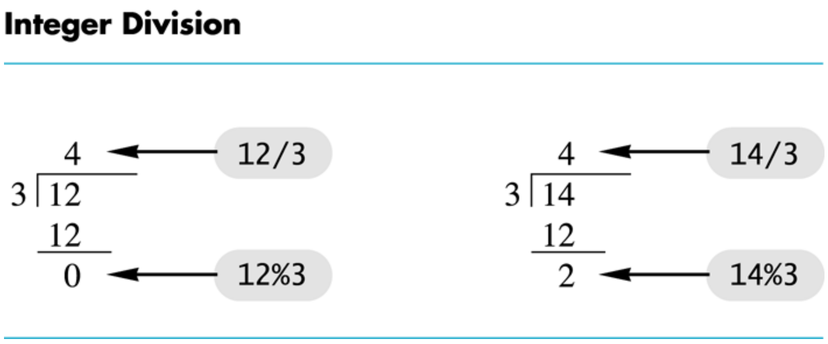
Fall 2019

Expressions

Common arithmetic operators

+ − * / %

- Can be used with any numeric type (integers and floating point numbers)
- Result of the operator depends on the type of the operands
- Be aware of the **integer division** (fractional part discarded)
  - 22/4 is 5



Integer Division

from: Problem Solving with C++, 10th Edition, Walter Savitch

“Rules”

- Use parentheses !
  - even when redundant
- Use whitespaces !

```
(b*b-4*a*c)/(2*a) 👎
```

```
((b * b) - (4 * a * c)) / (2 * a) 👍
```

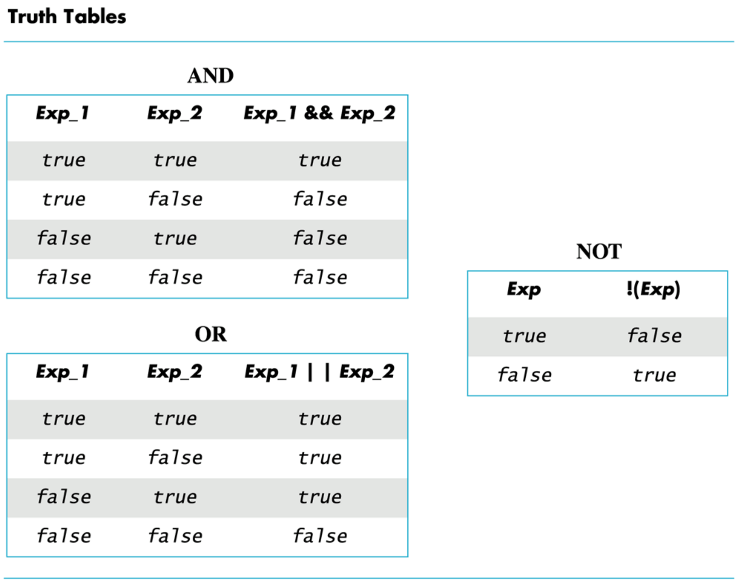
Boolean expressions

- Expressions that evaluate to either `true` or `false`
- Can use comparison operators

`==` `<` `>` `>=` `<=` `!=`

- Can use logical operators

`!` `&&` `||`



Truth Tables

AND

Exp_1	*Exp_2*	*Exp_1 && Exp_2*
true	*true*	*true*
true	*false*	*false*
false	*true*	*false*
false	*false*	*false*

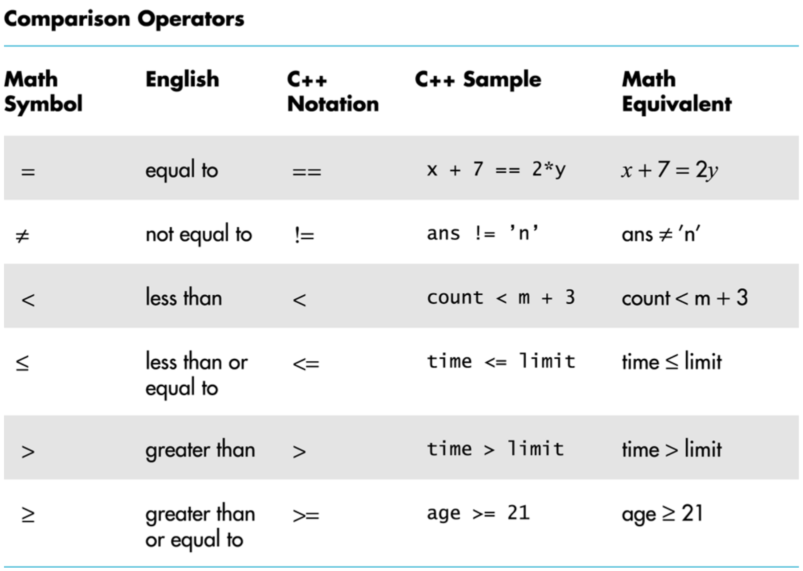
OR

Exp_1	*Exp_2*	*Exp_1 || Exp_2*
true	*true*	*true*
true	*false*	*true*
false	*true*	*true*
false	*false*	*false*

NOT

Exp	*!(Exp)*
true	*false*
false	*true*

from: Problem Solving with C++, 10th Edition, Walter Savitch



Comparison Operators

Math Symbol	English	C++ Notation	C++ Sample	Math Equivalent
=	equal to	`==`	`x + 7 == 2*y`	$x + 7 = 2y$
≠	not equal to	`!=`	`ans != 'n'`	ans ≠ 'n'
<	less than	`<`	`count < m + 3`	count < m + 3
≤	less than or equal to	`<=`	`time <= limit`	time ≤ limit
>	greater than	`>`	`time > limit`	time > limit
≥	greater than or equal to	`>=`	`age >= 21`	age ≥ 21

from: Problem Solving with C++, 10th Edition, Walter Savitch

Precedence Rules

The unary operators +, −, ++, −−, and !.
The binary arithmetic operations *, /, %
The binary arithmetic operations +, −
The Boolean operations <, >, <=, >=
The Boolean operations ==, !=
The Boolean operations &&
The Boolean operations ||

Highest precedence (done first)

Lowest precedence (done last)

from: Problem Solving with C++, 10th Edition, Walter Savitch

What is the value of this expression?

```
(x + 1) > 2 || (x + 1) < -3
```

Recommended style

```
((x + 1) > 2) || ((x + 1) < -3)
```

In C++ any non-zero value is **true** and zero is **false**

What is the value of this expression?

```
  false
(! 32 > 64)

(0 > 64)

false
```

Selection Statements
`if` and `switch`

`if` statements

- Allow conditional execution of code
- General idea:

```
if (expression)
    true statement
else
    false statement
```

The `if` statement (basic syntax)

```
if (expression)
    statementA


if (expression)
    statementA
else
    statementB
```

```
if (expressionA)
    statementA
else if (expressionB)
    statementB
  .
  .
  .
else
    statementN
```

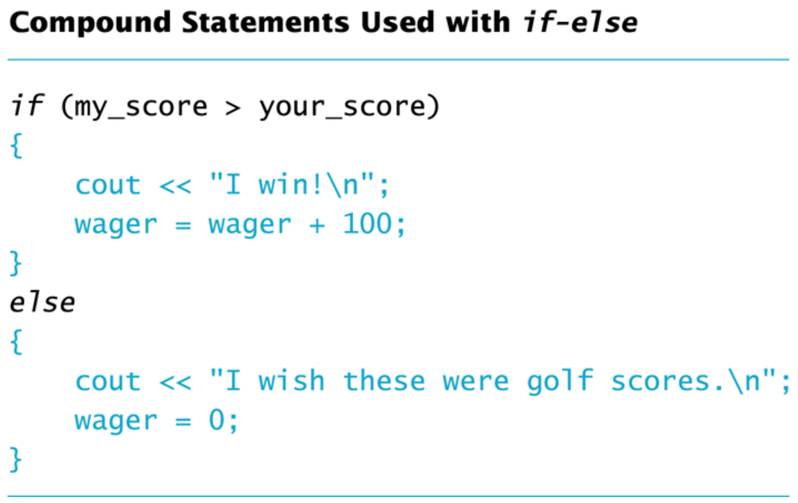
Example

```
int value;

std::cout << "Enter a number: ";
std::cin >> value;

if (value > 0) {
    std::cout << “positive number" << std::endl;
} else if (value < 0) {
    std::cout << "negative number" << std::endl;
} else {
    std::cout << "zero" << std::endl;
}
```

Compound statements

```
if (expression) {
    statementA
    statementB
    statementC
    ...
} else {
    statementL
    statementM
    statementN
    ...
}
```

- Recommended to **always use braces,** even with single statements
- Develop a good and consistent programming style



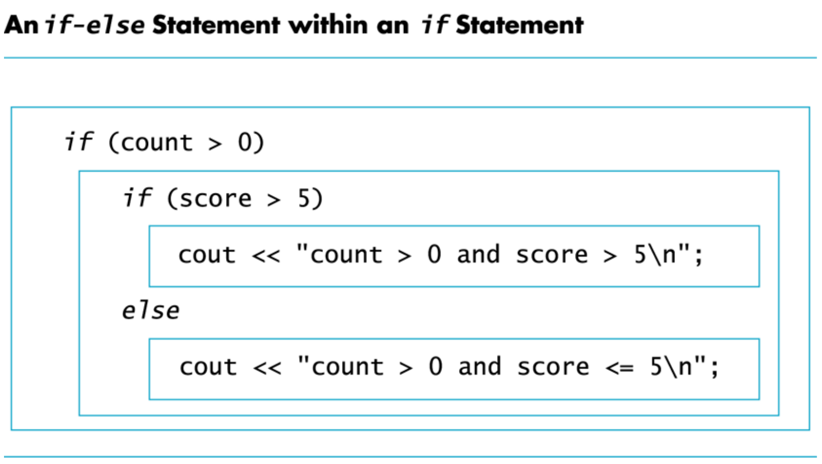
Compound Statements Used with *if-else*

```
if (my_score > your_score)
{
    cout << "I win!\n";
    wager = wager + 100;
}
else
{
    cout << "I wish these were golf scores.\n";
    wager = 0;
}
```

from: Problem Solving with C++, 10th Edition, Walter Savitch

Exercise

- Write a program in C++ (**on paper**) that:
  - reads the number of **hours**
  - calculates payment:
    - if number of hours no greater than 40, **payment** is calculated using the regular hourly rate of $35
    - if overtime, **payment** is calculated using the regular hourly rate for the first 40 hours and the special rate of $50 for the remaining hours
  - prints the calculated **payment**



An *if-else* Statement within an *if* Statement

```
if (count > 0)
    if (score > 5)
        cout << "count > 0 and score > 5\n";
    else
        cout << "count > 0 and score <= 5\n";
```

from: Problem Solving with C++, 10th Edition, Walter Savitch

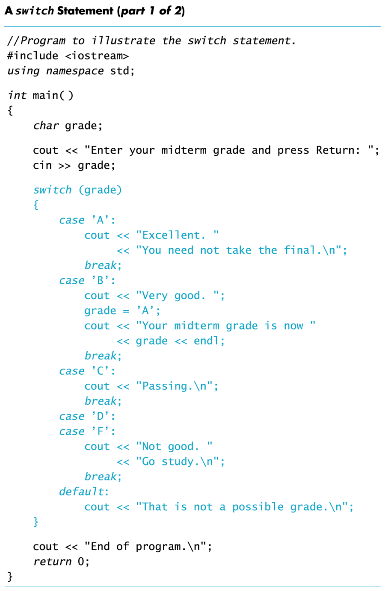
`switch` statements

- Allow conditional execution of code based on the value of an **integer** expression
- Basic syntax:

```
switch (expression) {
    case valueA:
        statementA
    case valueB:
        statementB
      .
      .
      .
    case valueN:
        statementN
    default:
        statement
}
```

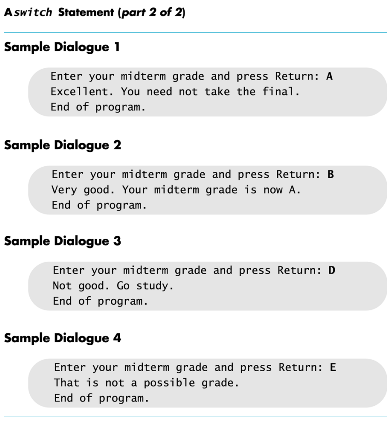
`if expression equals to a value, control executes corresponding statement (can be a compound statement), then continue executing statements until` **`break`** `is encountered`

characters (ascii values) can also be used in switch statements



A *switch* Statement (*part 1 of 2*)

```
//Program to illustrate the switch statement.
#include <iostream>
using namespace std;

int main( )
{
    char grade;

    cout << "Enter your midterm grade and press Return: ";
    cin >> grade;

    switch (grade)
    {
        case 'A':
            cout << "Excellent. "
                 << "You need not take the final.\n";
            break;
        case 'B':
            cout << "Very good. ";
            grade = 'A';
            cout << "Your midterm grade is now "
                 << grade << endl;
            break;
        case 'C':
            cout << "Passing.\n";
            break;
        case 'D':
        case 'F':
            cout << "Not good. "
                 << "Go study.\n";
            break;
        default:
            cout << "That is not a possible grade.\n";
    }

    cout << "End of program.\n";
    return 0;
}
```



A *switch* Statement (*part 2 of 2*)

Sample Dialogue 1

```
Enter your midterm grade and press Return: A
Excellent. You need not take the final.
End of program.
```

Sample Dialogue 2

```
Enter your midterm grade and press Return: B
Very good. Your midterm grade is now A.
End of program.
```

Sample Dialogue 3

```
Enter your midterm grade and press Return: D
Not good. Go study.
End of program.
```

Sample Dialogue 4

```
Enter your midterm grade and press Return: E
That is not a possible grade.
End of program.
```

from: Problem Solving with C++, 10th Edition, Walter Savitch