

Lecture #4

Selections

chapter: 1.4.4, 5.3,5.3.1,5.3.2

Motivations

If you assigned a negative value for radius in earlier example, ComputeArea.cpp, the program would print an invalid result. If the radius is negative, you don't want the program to compute the area. How can you deal with this situation?

Objectives

- To declare **bool** variables and write Boolean expressions using relational operators.
- To implement selection control using one-way/two-way **if** statements.
- To implement selection control using nested **if** and multi-way **if-else** statements.
- To avoid common errors and pitfalls in **if** statements.
- To program using selection statements with combined conditions (**LeapYear**, **Lottery**).
- To implement selection control using **switch** statements.
- To write expressions using the conditional operator.
- To implement enum types

The bool Type and Operators

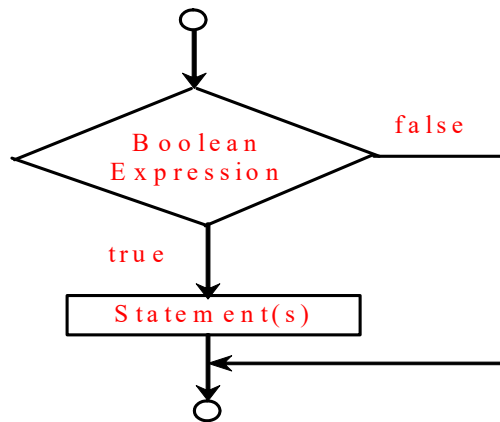
Often in a program you need to compare two values, such as whether *i* is greater than *j*. C++ provides six *relational operators* (also known as *comparison operators*) in next slide that can be used to compare two values.

Relational Operators

Operator	Name	Example	Result
<	less than	1 < 2	true
<=	less than or equal to	1 <= 2	true
>	greater than	1 > 2	false
>=	greater than or equal to	1 >= 2	false
==	equal to	1 == 2	false
!=	not equal to	1 != 2	true

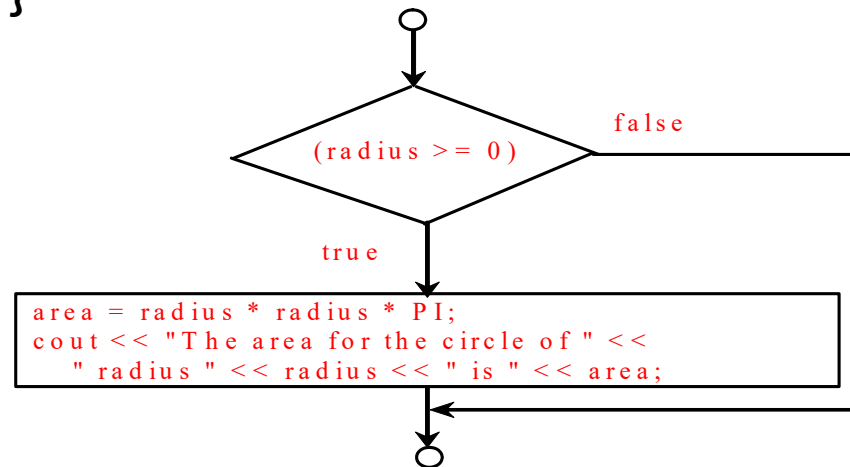
One-way if Statements

```
if (booleanExpression)
{
    statement(s);
}
```



(a)

```
if (radius >= 0)
{
    area = radius * radius * PI;
    cout << "The area for the circle of " <<
        " radius " << radius << " is " << area;
}
```



(b)

Note

Outer parentheses

```
if ((i > 0) && (i < 10))  
{  
    cout << "i is an " <<  
        "integer between 0 and 10";  
}
```

(a)

Equivalent

Braces can be omitted if the block contains a single statement

```
if ((i > 0) && (i < 10))  
    cout << "i is an " <<  
        "integer between 0 and 10";
```

(b)

Examples

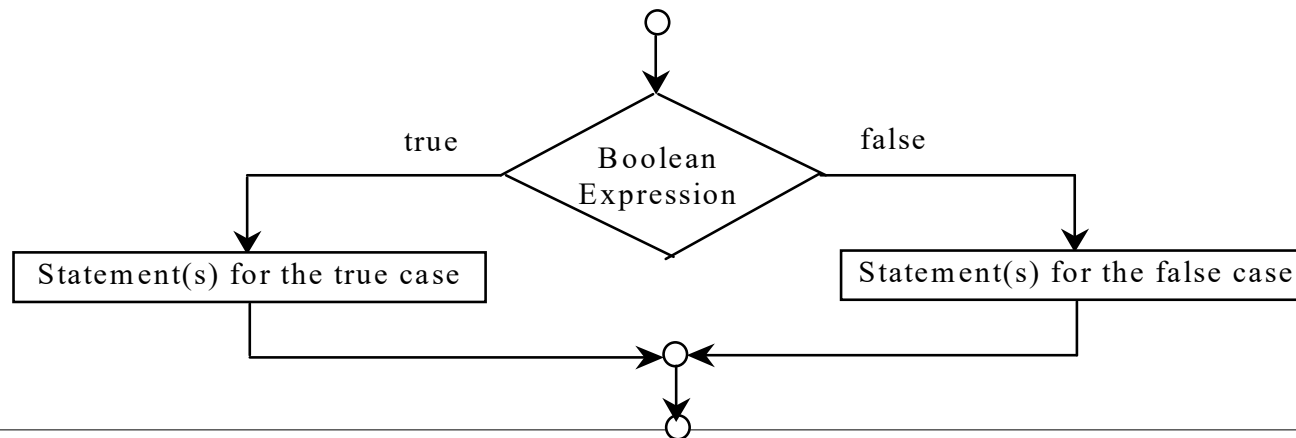
Below is a program that checks whether a number is even or odd. The program prompts the user to enter an integer and displays “number is even” if it is even and “number is odd” if it is odd.

```
int number;  
cout << "Enter an integer: ";  
cin >> number;  
  
if (number % 2 == 0)  
{  
    cout << number << " is even.";  
}  
  
if (number % 2 != 0)  
{  
    cout << number << " is odd.";  
}
```

[TestBoolean](#)

The if...else Statement

```
if (booleanExpression)
{
    statement(s)-for-the-true-case;
}
else
{
    statement(s)-for-the-false-case;
}
```



Revisting TestBoolean

```
int number;  
  
cout << "Enter an integer: ";  
cin >> number;  
  
if (number % 2 == 0)  
{  
    cout << number << " is even.";  
}  
  
else  
{  
    cout << number << " is odd.";  
}
```

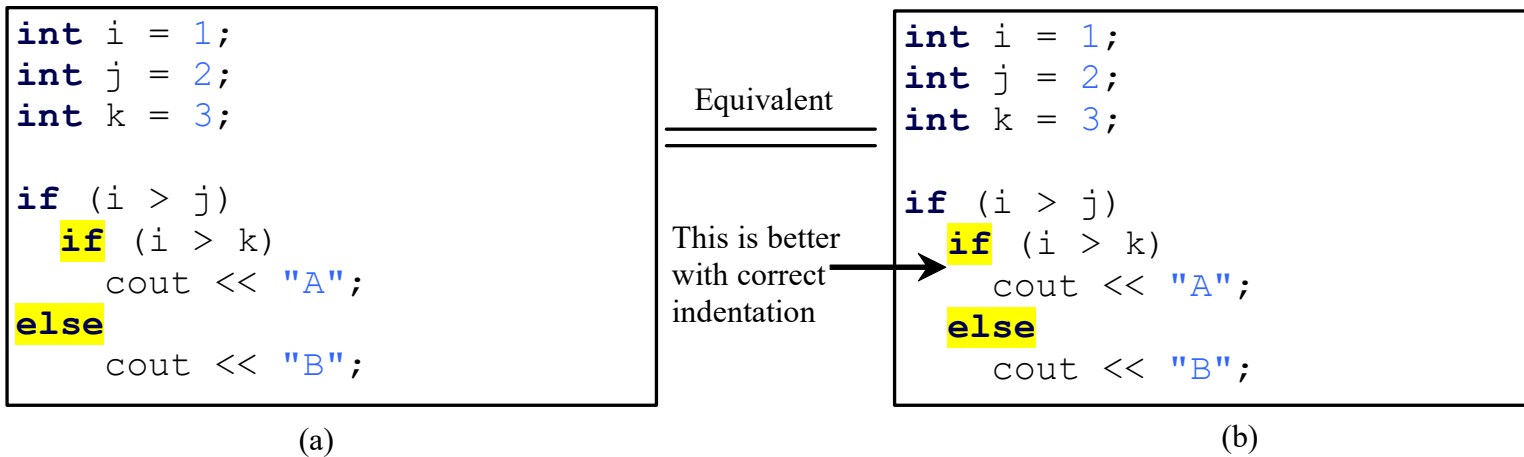
[TestBoolean](#)

Nested if Statements

```
if (i > k)
{
    if (j > k)
        cout << "i and j are greater than k";
}
else
    cout << "i is less than or equal to k";
```

Note

The else clause matches the most recent if clause in the same block.



Note, cont.

Nothing is printed from the preceding statement. To force the else clause to match the first if clause, you must add a pair of braces:

```
int i = 1; int j = 2; int k = 3;

if (i > j)
{
    if (i > k)
        cout << "A";
}
else
    cout << "B";
```

This statement prints B.

Example: A Simple Math Learning Tool

This example creates a program for a first grader to practice subtractions. The program randomly generates two single-digit integers number1 and number2 with number1 \geq number2 and displays a question such as “What is $9 - 2$?” to the student, as shown in the sample output. After the student types the answer, the program displays a message to indicate whether the answer is correct.

[SubtractionQuiz](#)

Examples

This example shows a program that checks whether a number is divisible by 2 and 3, whether a number is divisible by 2 or 3, and whether a number is divisible by 2 or 3 but not both:

TestBooleanOperators

```
int number;
cout << "Enter an integer: ";
cin >> number;

if (number % 2 == 0 and number % 3 == 0)
    cout << number << " is divisible by 2 and 3." << endl;

if (number % 2 == 0 or number % 3 == 0)
    cout << number << " is divisible by 2 or 3." << endl;

if ((number % 2 == 0 or number % 3 == 0) && !(number % 2 == 0 and number % 3 == 0))
    cout << number << " divisible by 2 or 3, but not both." << endl;
```

Exercise: Do in class

Write a program that lets the user enter a year and checks whether it is a leap year.

A year is a *leap year* if it is divisible by 4 but not by 100 or if it is divisible by 400.

LeapYear

switch Statements

switch (status)

{

case 0: compute taxes for single filers;

break;

case 1: compute taxes for married file jointly;

break;

case 2: compute taxes for married file separately;

break;

case 3: compute taxes for head of household;

break;

default: cout << "Errors: invalid status" << endl;

}

Switch statement

Suppose day is 3:

```
switch (day)
{
    case 1: // Fall to through to the next case
    case 2: // Fall to through to the next case
    case 3: // Fall to through to the next case
    case 4: // Fall to through to the next case
    case 5: cout << "Weekday"; break;
    case 0: // Fall to through to the next case
    case 6: cout << "Weekend";
}
```

Enumerated Types

```
enum Day {MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY};
```

Once a type is defined, you can declare a variable of that type:

```
Day day;
```

The variable day can hold one of the values defined in the enumerated type. For example, the following statement assigns enumerated value MONDAY to variable day:

```
day = MONDAY;
```

[TestEnumeratedType](#)

Enumerated Types

in C++11: strongly typed enum

// This code WILL compile in C++11 up

```
enum class Color {RED, GREEN, BLUE};
enum class Feelings {EXCITED, MOODY, BLUE};

Color r = Color::RED;
Feelings f = Feelings::EXCITED;

if (f == Feelings::EXCITED && r == Color::RED)
    cout << "It' normal to be excited and red!" << endl;
else
    cout << "This is not expected result!" << endl;
```

Enum.cpp

Exercise: Do it at home



Mittuniversitetet
MID SWEDEN UNIVERSITY

- Convert the following **if-else** statement **into** corresponding **Switch-statement**:

```
if (t == 3)
```

```
    x = 30;
```

```
else
```

```
    if (t == 4)
```

```
        x = 10;
```

```
    else
```

```
        if (t == 2 || t == 7 || t == 8)
```

```
            x = 20;
```

```
        else
```

```
            x = 100;
```

Thank you