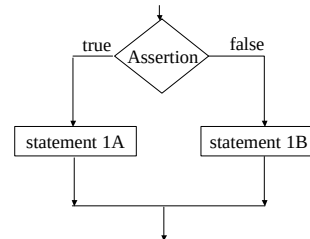


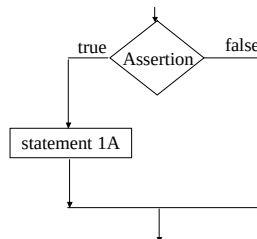
Flow of Control

- Flow of control is the order in which the computer executes statements in a program.
- Flow of control is normally sequential.
- Control structures allow for non-sequential flow of control.
- The first control structure we are going to look at is a selection control structure.

Selection



More Selection



The `if` Statement

- The If-Then-Else Form
IfStatement (the If-Then-Else form)
`if` (Expression)
 Statement1A
`else`
 Statement1B

The `if` Statement

- The If-Then-Else Form
IfStatement (the If-Then-Else form)
`if` (Expression)
 Statement1A
`else`
 Statement1B
- The If-Then Form
IfStatement (the If-Then form)
`if` (Expression)
 Statement1A

Conditions and Logical Expressions

- Asking questions in C++ means making an assertion that is either true or false.
- The computer evaluates the assertion.
- We have a new data type that is used in assertions, `bool`. It has two possible values `true` and `false`.
- We declare variables of type `bool` in the same way we declare variables of other types.
 - Example `bool dataOK;`

Logical Expressions

- Assertions take the form of logical expressions, they evaluate to either **true** or **false**.
- Examples
 - Boolean variable or constant
 - An expression followed by a relational operator followed by an expression.
 - A logical expression followed by a logical operator followed by a logical expression.

Relational Operators

- Relational Operators test the relationship between two values.
- Example `bool lessThan; int i, j;`
`cin >> i >> j;`
`lessThan = (i < j);`
- Operators

<code>==</code>	Equal to
<code>!=</code>	Not equal to
<code>></code>	Greater than
<code><</code>	Less than
<code>>=</code>	Greater than or equal to
<code><=</code>	Less than or equal to

Relational Operator Examples

x	y	Expression	Result
12	2	<code>x + 3 <= y * 10</code>	true

- Caution: The relational operator, `==`, and the assignment operator, `=`, are not the same.

Relational Operator Examples

x	y	Expression	Result
12	2	<code>x + 3 <= y * 10</code>	true
20	2	<code>x + 3 <= y * 10</code>	false

- Caution: The relational operator, `==`, and the assignment operator, `=`, are not the same.

Relational Operator Examples

x	y	Expression	Result
12	2	<code>x + 3 <= y * 10</code>	true
20	2	<code>x + 3 <= y * 10</code>	false
7	1	<code>x + 3 != y * 10</code>	false

- Caution: The relational operator, `==`, and the assignment operator, `=`, are not the same.

Relational Operator Examples

x	y	Expression	Result
12	2	<code>x + 3 <= y * 10</code>	true
20	2	<code>x + 3 <= y * 10</code>	false
7	1	<code>x + 3 != y * 10</code>	false
17	2	<code>x + 3 == y * 10</code>	true

- Caution: The relational operator, `==`, and the assignment operator, `=`, are not the same.

Relational Operator Examples

x	y	Expression	Result
12	2	$x + 3 \leq y * 10$	true
20	2	$x + 3 \leq y * 10$	false
7	1	$x + 3 \neq y * 10$	false
17	2	$x + 3 \neq y * 10$	true
100	5	$x + 3 > y * 10$	true

- Caution: The relational operator, `==`, and the assignment operator, `=`, are not the same.

Logical Operators

- The binary AND operator (`&&`) requires both relationships to be true in order for the overall result to be true.
- The binary OR operator (`||`) has an overall result of true if either or both of two logical expressions are true.
- The unary NOT operator (`!`) has a result of true if the logical expression is false.

Precedence of Operators Revisited

! Unary + Unary - Highest precedence
 / % *
 + -
 < <= > >=
 == !=
 &&
 ||
 = Lowest precedence

Warning

- Do not compare floating-point numbers for equality. Remember that the binary representations of floating-point numbers are approximations. Instead, test for near equality.
 – Example: `fabs(r - s) < 0.00001`

if Statement Examples (If-Then-Else Form)

```
if (hours <= 40.0)
    pay = rate * hours;
else
    pay = rate * (40.0 + (hours - 40.0) * 1.5);
cout << pay;
```

if Statement Examples (If-Then-Else Form)

```
if (hours <= 40.0)
    pay = rate * hours;
else
    pay = rate * (40.0 + (hours - 40.0) * 1.5);
cout << pay;

if (divisor != 0)
    result = dividend / divisor;
else
    cout << "Division by zero is not allowed. " << endl;
```

Block Statements

- Anytime one statement can be used, a series of statements can be used, they must be enclosed in braces {}.

- Example:

```
if (divisor != 0)
{
    result = dividend / divisor;
    cout << "Division performed." << endl;
}
else
{
    cout << "Division by zero is not allowed." << endl;
    result = 9999;
}
```

if Statement Examples (If-Then Form)

```
if (age < 18)
    cout << "Not an eligible";
cout << "voter." << endl;
```

if Statement Examples (If-Then Form)

```
if (age < 18)
    cout << "Not an eligible";
cout << "voter." << endl;

if (result < 0.0)
{
    cout << "Check box 24A" << endl;
    result = 0.0;
}
```