Compound Assignment Operators in C++

Simple assignment operator	Compound assignment operator
X = X + 1;	X += 1;
Y = Y – 1 ;	Y -= 1 ;
Z = Z + X ;	Z += X ;
P = P * item ;	P *= item ;
N = N * (x + 1) ;	N *= (x + 1) ;
Total = Total / (X+Y);	Total /= (X+Y);
Hours = Hours % 13;	Hours %= 13;

Control Structures (Selections)

Topics to cover here:

Selection statements in the algorithmic language:

- One-Way Selection
- Two-Way Selection
- Multi-Way Selection
- Nested Structures

- Selection statements in C++ language

Selections .. cont.

- The selection statement allows to choose a set of statements for execution.
- The selection depends on the validity of a ***condition*** when the program is executed.
- The ***condition*** is a logical expression that has values as ***true*** or ***false***.

One-Way Selection

Syntax:

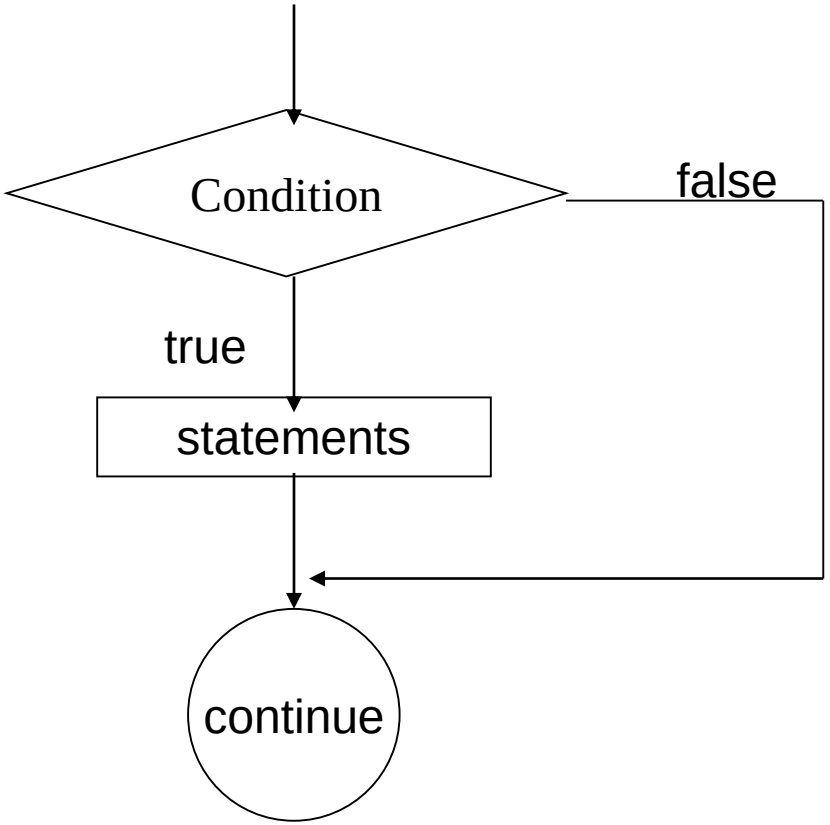
In pseudo code	In C++
IF (*condition)* **THEN** *statements* **END IF**	**if** (logical expression) statements;

The semantics (execution) of this statement:

- If the value of the *"condition"* is **true**, the statements between IF .. END IF are executed and the execution continues to the statement after END IF.
- If the value of the *"condition"* is **false**, the statements between IF .. END IF are ignored and the execution continues from the statement that follows END IF.

One-Way Selection

The following figure shows the execution of this selection statement.



One-Way Selection .. Examples

- **Example 1:**

Write an algorithm that takes an integer and prints its double value if it is less than 50.

First, we have to analyze the problem to understand what is its requirements and how to solve it.

1- Analysis stage:

- ***Problem Input:***

  - An integer, say n

- ***Problem Output:***

  - The double value of n

- ***Criteria***

  if n < 50, print its double value

Example 1 .. cont.

2- Algorithm Design

ALGORITHM Double

INPUT n

IF (n < 50) **THEN**

OUTPUT "The double value is " , n * 2

END IF

OUTPUT " Finished"

END

Example 1 .. cont.

3- Testing the algorithm

n	(n < 50)
12	---
	true

The output:

The double value is 24

Finished

Example 1: C++ Program

```
#include <iostream>
using namespace std;
void main ( )
{
   int   n ;
   cin >> n ;
   if (n < 50)
        cout <<    “The double value is “ << n * 2 << endl ;
   cout << “ Finished “ << endl;
}
```

Two-Way Selection

This statement chooses- statements to run- from two sets of choices.

Syntax:

In pseudo code	In C++
IF (*condition*) **THEN** *statements1* **ELSE** *statements2* **END IF**	**if** (logical expression) statements1; **else** statements2;

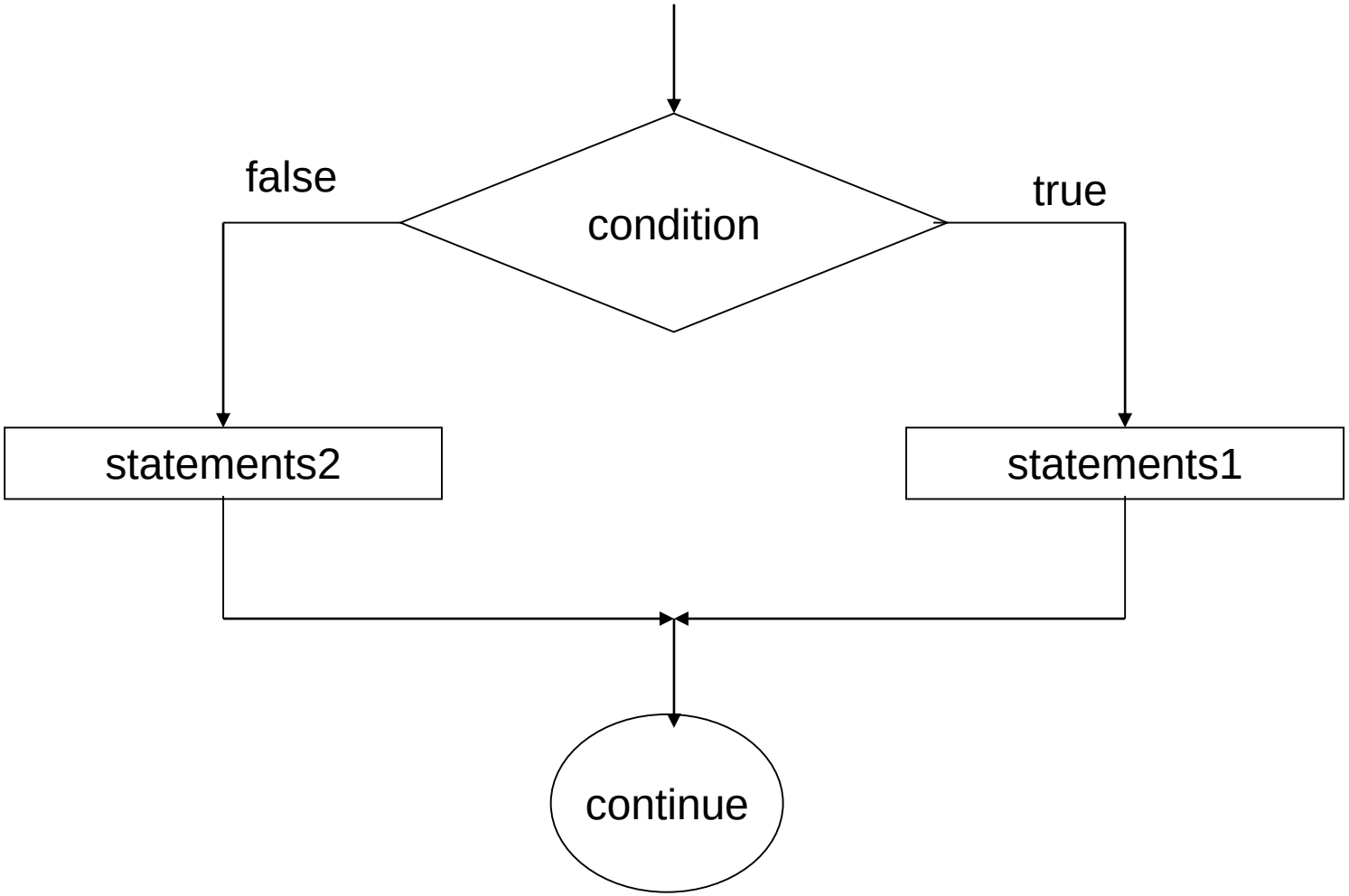
Two-Way Selection .. cont.

- **The semantics (execution) of this statement**:

- If the value of the *"condition"* is **true**, the statements after THEN are executed and the execution continues to the statement after END IF.
- If the value of the *"condition"* is **false**, the statements after ELSE are executed and the execution continues from the statement that follows END IF.

- The following figure shows this execution:

Two-Way Selection .. cont.



Two-Way Selection .. Examples

- **Example 1:**

  Write an algorithm that takes two integers and prints the smallest number with appropriate message.

1- Analysis stage:

- ***Problem Input:***

  - Two integers, say num1 and num2

- ***Problem Output:***

  - The smaller number

Example 1 .. cont.

2- Algorithm Design

ALGORITHM Smaller

INPUT num1, num2

IF (num1 < num2) **THEN**

OUTPUT “The smaller number is “ , num1

ELSE

OUTPUT “The smaller number is “ , num2

END IF

END Smaller

Example 1 .. cont.

3- Testing the algorithm

num1	num2	(num2<num1)
9	3	---
		true

The output:

The smaller number is 3

Example 1: C++ Program

```
#include <iostream>
using namespace std;
void main ( )
{  int   num1, num2 ;
   cin >> num1 >> num2 ;
   if ( num1 < num2 )
       cout <<    “The smaller value is  “ <<  num1 << endl ;
   else
      cout <<    “The smaller value is  “ <<  num12 << endl ;
}
```

Example 2

Write an algorithm that takes prices of two items, if there total is over 100 JDs, calculate a discount of 40% and tax of 16%. Otherwise, calculate a discount of 20% and tax of 6%. Then calculate the total price as: Total + tax – Discount.

1- Analysis stage*:*

- ***Problem Input:***

  - Two prices, p1 and p2

- ***Problem Output:***

- The total price of the two items

- ***Criteria***

check whither the total prices is over 100 or not.

Example 2 .. cont.

2- Algorithm Design

```
ALGORITHM    Total_Price
   INPUT   p1, p2
   sub_total ← p1 + p2
   IF (sub_total > 100) THEN
       Discount ← sub_total * 0.4
       Tax ← sub_total * 0.16
   ELSE
       Discount ← sub_total * 0.2
       Tax ← sub_total * 0.06
   END IF
   Total ← sub_total + Tax - Discount
 END  Total_Price
```

Example 2: C++ Program

```
#include <iostream>
using namespace std;
void main ( )
{  float   p1, p2, sub_total, discount, tax, total;
   cin >> p1 >> p2;
   sub_total = p1 + p2;
   if ( sub_total > 100)
   {
      discount = sub_total * 0.4;
      tax = sub_total * 0.16;
    }
   else
   {
      discount = sub_total * 0.2;
      tax = sub_total * 0.06;
   }
}
```

Multi Way Selection

- You can choose statement(s) to run from many sets of choices.
- There are two cases for this:

(a) Multi way selection by nested IF structure

(b) Multi way selection by SWITCH structure

Multi Way Selection by Nested IF Structure

- The structure that contains another structure of the same type is called a nested structure.

- In the Nested IF structure, the statements that exists between IF and ELSE or between IF and END IF can contain IF statement.

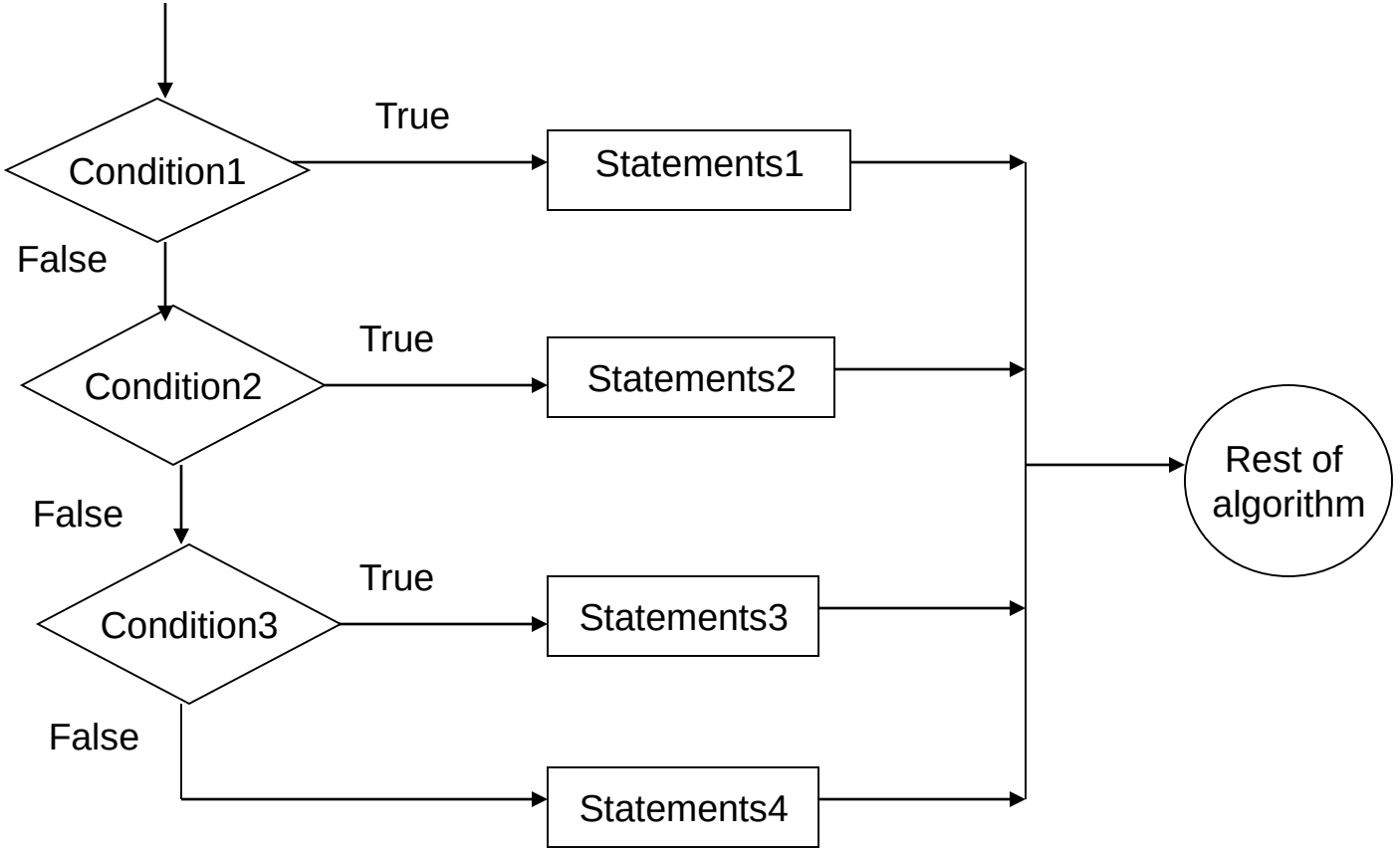
Multi Way Selection by Nested If Structure .. Cont.

Syntax:

In pseudo code	In C++
IF (*condition1*) **THEN** *Statements1* **ELSE IF** (*condition2*) **THEN** *Statements2* **ELSE IF** (*Condition3*) **THEN** *Statements3* **ELSE IF** (*Condition4*) **THEN** *Statements4* **END IF** **END IF** **END IF**	**if** (*condition1*) *Statements1;* **else if** (*condition2*) *Statements2 ;* **else if**(*Condition3*) *Statements3;* **else if**(*Condition4*) *Statements4;*

Note: The nest can be to many levels.

Multi Way Selection by Nested If Structure .. Cont.



Multi Way Selection by Nested If Structure .. Examples

- **Example 1:**

  Write an algorithm that inputs a student mark and outputs the corresponding grade, where grades are as follows:

mark	grade
90-100	A
80-89	B
70-79	C
60-69	D
< 60	E

Example 1 .. Cont.

1- Analysis stage*:*

- ***Problem Input:***

  - student's mark, mark

- ***Problem Output:***

  - grade

- ***Criteria***

  - according to the previous grade table

Example 1 .. Cont.

2- Algorithm Design

```
ALGORITHM   Grades
   INPUT   mark
   IF ( mark < 0   OR   mark > 100 ) THEN
      OUTPUT " Mark out of range"
   ELSE IF ( mark ≥ 90 AND mark ≤ 100 ) THEN
               OUTPUT   "A"
          ELSE IF ( mark ≥ 80 ) THEN
                     OUTPUT   "B"
                ELSE IF ( mark ≥ 70 ) THEN
                           OUTPUT   "C"
                      ELSE IF ( mark ≥ 60 ) THEN
                                 OUTPUT   "D"
                            ELSE       OUTPUT   "E"
                                 END IF
                      END IF
                END IF
          END IF
    END IF
END  Grades
```

Example 1: C++ Program

```
#include <iostream>
using namespace std;
void main ( )
{ int mark ;
  cin >>   mark;
    if ( mark < 0  ||   mark > 100 )
         cout <<  " Mark out of range" << endl;
   else if ( mark >= 90  )
           cout   <<  "A" << endl ;
         else if ( mark >= 80 )
                  cout   <<  "B" << endl ;
              else if ( mark >= 70 )
                       cout   <<  "C" << endl ;
                   else if ( mark >= 60 )
                            cout   <<  "D" << endl ;
                        else     cout   <<  "E" << endl ;
}
```

Example 2: : Read three numbers to print the smallest one.

```
#include <iostream.h>
void main() {
        int a, b, c;
        cout<<"\nPlease Enter three numbers:";
        cin>>a>>b>>c;
        cout<<"\nMin= ";
        if ((a < b) && (a < c))
                cout<<a;
        if ((b < a) && (b < c))
                cout<<b;
        if ((c < a) && (c < b))
                cout<<c;
        cout<<endl;
}
```

Program2 (nested if)

```
#include <iostream.h>
void main() {
        int a, b, c;
        cout<<"\nPlease Enter three numbers:";
        cin>>a>>b>>c;
        cout<<"\nMin= ";
        if (a < b)
                if (a < c)
                        cout<<a;
                else
                        cout<<c;
        else
                if (b < c)
                        cout<<b;
                else
                        cout<<c;
        cout<<endl;
}
```

Note:

- In some cases, you may need to use a variable as the condition of an if statement. But, how this variable will be interpreted into the result of the condition?

We can use: If(x)

- **Example:**

```
{
   int x;
   cin>> x;
   if (x)
     cout << “x is any number but zero”;
   else
     cout << “ x is zero”;
}
```

Notes: - a value of 0, is considered false in this case, while all other values, either positive or negative, are considered true.

- The same applies for float variables.
- Character variables always return true.

Switch Statement in C++

Switch Statement in C++

- **Syntax**

```
switch (selector)
{ case L1: statements1; break;
  case L2: statements2; break;
   …
  default: statements_n;
 }
```

- **Semantics:**

This statement has the same meaning as in the algorithmic language.

Example 1: C++ Program

```
#include <iostream>
using namespace std;
void main ( )
{
   int lab;
   cin >> lab;
   switch  ( lab )
   {
      case 503 : cout <<  " C++ " << endl; break;
      case 508: cout <<  " C# " << endl; break;
     case 512 : cout <<  " Oracle " << endl; break;
    case 514: cout <<  " PHP " << endl; break;
     case 507: cout <<  " Java " << endl; break;
     default :  cout << " MS Office " << endl;
   }
}
```

Example 2: C++ Program

```
#include <iostream>
using namespace std;
void main ( )
{ char ch ;
   cout << " \n Enter the grade of student: "<<endl ;
   cin >>   ch;
 switch (ch) {
case  'A' :
case   'a' :
              cout<<"Excellent";
               break;
case  'B' :
case   'b' :
              cout<<"Good";
              break;
case  'C' :
case   'c' :
              cout<<"O.K";
              break;
case  'D' :
case   'd' :
case  'F' :
case   'f' :
              cout<<"poor";
              break;
default:    cout<<"invalid letter grade"; }
}
```

Example 3: C++ Program

```
#include <iostream>
using namespace std;
void main()
{ int x,y;
 cout << "Enter 2 integer number: ";
 cin >> x>>y;
 switch (x+y)
 {  case 7: cout << "Too small, sorry!";
                break;
    case 5: cout << "Good job!\n";
                break;
    case 4: cout << "Nice Pick!\n";
    case 3: cout << "Excellent!\n";
               break;
    case 2: cout << "Masterful!\n";
              break;
    case 1: cout << "Incredible!\n";
                break;
    default: cout << "Too large!\n";
 }
cout << "\n\n";
 }
```

Example 4: C++ Program

```
#include <iostream>
using namespace std;
void main ( )
{ char ch ;    float radius, area, circum;
  cout << " Enter the radius of a circle: " ;
  cin >>   radius;
  cout << " Enter a to calculate the area of a circle or c to calculate its circumference:"
  cin >> ch ;
  switch (ch)
   { case 'a' :  area = 3.14f * radius * radius;
                 cout <<  " Area = " << area << endl;  break;
     case'c' :   circum = 2 * radius * 3.14f ;
                 cout << " Circumference =  " << circum << endl; break;
     default :   cout << " Invalid letter was read " << endl;
   }
}
```

Convert IF into Switch

```
#include <iostream>
using namespace std;
void main ( )
{ char degree;
   cout << " \n Enter the temperature degree
     "<<endl ;
   cin >>   degree;
   if (degree >= 50)
     cout<<"Very Hot";
   else if (degree >= 35 && degree <50)
          cout<<"Hot";
        else if (degree >= 20 && degree <35)
               cout<<"Fair";
             else if (degree >= 0 && degree
     <20)
                    cout<<"Cold";
                else
                     cout<<"Very Cold";

}
```

```
#include <iostream>
using namespace std;
void main ( )
{ char degree;
   cout << " \n Enter the temperature degree "<<endl ;
   cin >>   degree;
   switch (degree >= 50)
   {
     case true:  cout<<"Very Hot";  break;
     case false:
       switch (degree >= 35 && degree <50)
         {
          case true: cout<<"Hot";  break;
         case false:
               switch (degree >= 20 && degree <35)
                    {
                     case true: cout<<"Fair"; break;
                     case false:
                     switch (degree >= 0 && degree <20)
                               {
                          case true: cout<<"Cold";  break;
                          case false:  cout<<"Very Cold";
                               }
                    }
          }  }  }
```

Increment and Decrement Operators in C++

Increment Operators	Decrement Operators
1- Postfix operator: e.g. i++ 2- Prefix operator: e.g. ++i	1- Postfix operator: e.g. i-- 2- Prefix operator: e.g. --i

- For postfix operators, the increment (or decrement) occurs after the current value of the variable has been used.
- For prefix operators, the increment (or decrement) occurs first and the new value of the variable is then used.
- **Example**

The following C++ statements has the effects as shown in the comment:

```
i = 3;          // initial value of i
k = i++;        // assigns 3 to k and 4 to i
k = ++i;        // assigns 5 to k and 5 to i
k = i-- ;       // assigns 5 to k and 4 to i
k = --i ;       // assigns 3 to k and 3 to i
```

Increment and Decrement Operators

Simple	easy	easiest
x = x+1;	x += 1	x++
x = x-1;	x -= 1	x--

Example

- Compute the following expression where: z=3, x=2, y=2

  Z+= ++x - y--;

Solution: