

CHAPTER 2.1

CONTROL STRUCTURES (SELECTION)

Dr. Shady Yehia Elmashad



Outline

- 1. Introduction**
- 2. Algorithms**
- 3. Pseudocode**
- 4. Control Structures**
- 5. The if Selection Structure**
- 6. The if/else Selection Structure**
- 7. The switch Multiple-Selection Structure**

1. Introduction

- Before writing a program:
 - Have a thorough understanding of problem
 - Carefully plan your approach for solving it
- While writing a program:
 - Know what “building blocks” are available
 - Use good programming principles

2. Algorithms

- All computing problems
 - can be solved by executing a series of actions in a specific order
- Algorithm
 - A procedure determining the
 - Actions to be executed
 - Order in which these actions are to be executed
- Program control
 - Specifies the order in which statements are to be executed

3. Pseudocode

- Pseudocode

- Artificial, informal language used to develop algorithms
- Similar to everyday English
- Not actually executed on computers
- Allows us to “think out” a program before writing the code for it
- Easy to convert into a corresponding C++ program
- Consists only of executable statements

4. Control Structures

- Sequential execution
 - Statements executed one after the other in the order written
- Transfer of control
 - When the next statement executed is not the next one in sequence
- Bohm and Jacopini: all programs written in terms of 3 control structures
 - Sequence structure
 - Built into C++. Programs executed sequentially by default.
 - Selection structures
 - C++ has three types - **if**, **if/else**, and **switch**
 - Repetition structures
 - C++ has three types - **while**, **do/while**, and **for**

4. Control Structures

- Flowchart

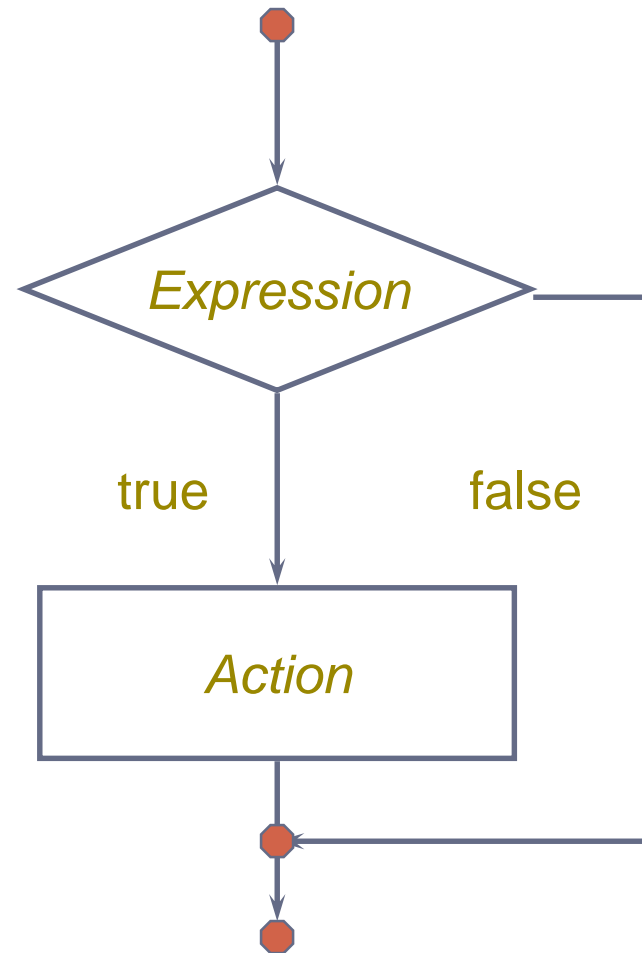
- Graphical representation of an algorithm
- Drawn using certain special-purpose symbols connected by arrows called flowlines.
- Rectangle symbol (action symbol)
 - Indicates any type of action.
- Oval symbol
 - indicates beginning or end of a program, or a section of code (circles).

- single-entry/single-exit control structures

- Connect exit point of one control structure to entry point of the next (control-structure stacking).
- Makes programs easy to build.

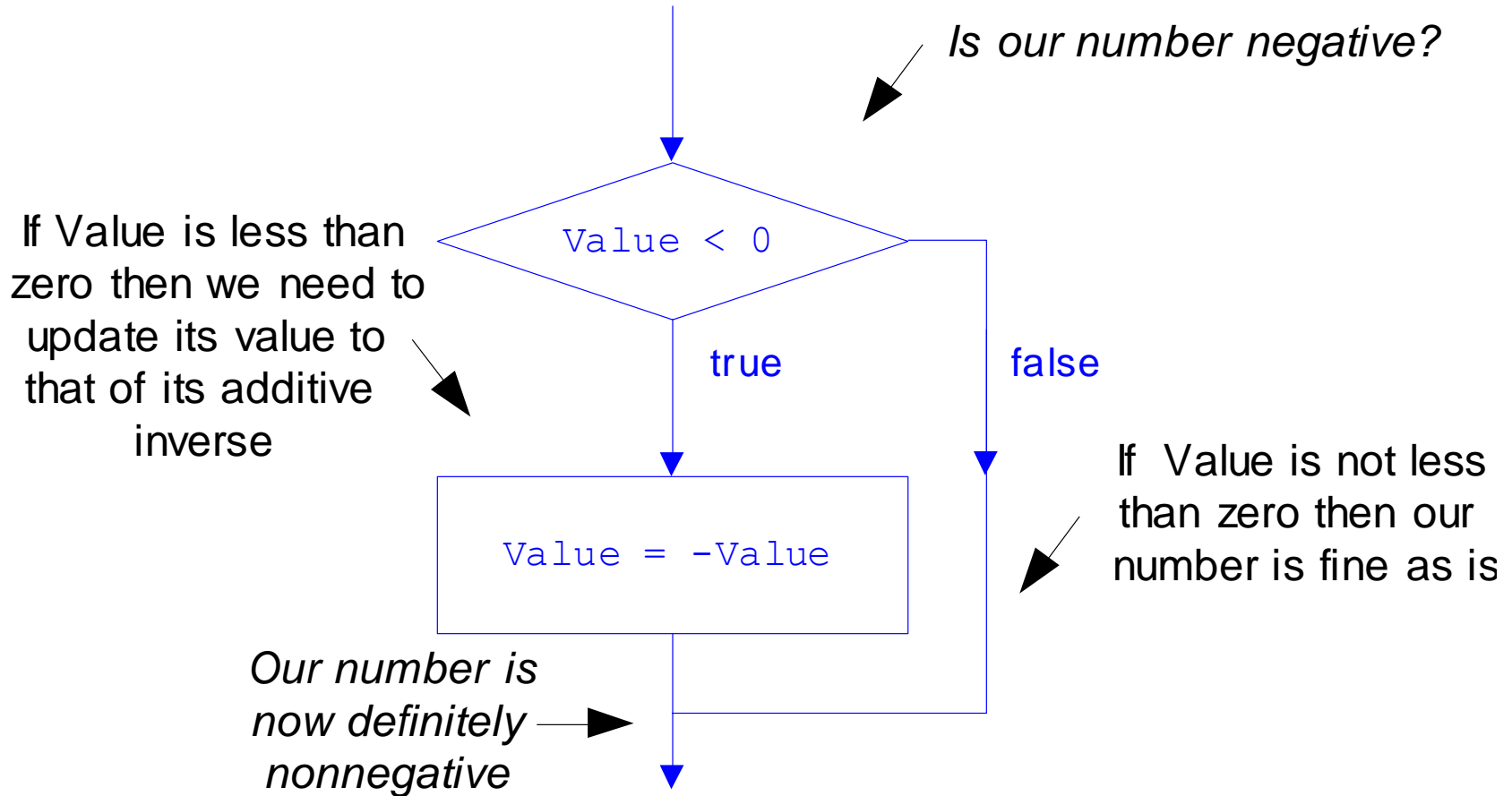
5. The if Selection Structure

- used to choose among alternative courses of action.
- Syntax
if (*Expression*)
Action
- If the *Expression* is true then execute *Action*.
- *Action* is either a single statement or a group of statements within braces.



5. The if Selection Structure

Example



5. The if Selection Structure

- Pseudocode example:

*If student's grade is greater than or equal to 60
Print "Passed"*

- If the condition is **true**
 - print statement executed and program goes on to next statement
- If the condition is **false**
 - print statement is ignored and the program goes onto the next statement
- Indenting makes programs easier to read
 - C++ ignores whitespace characters

5. The if Selection Structure

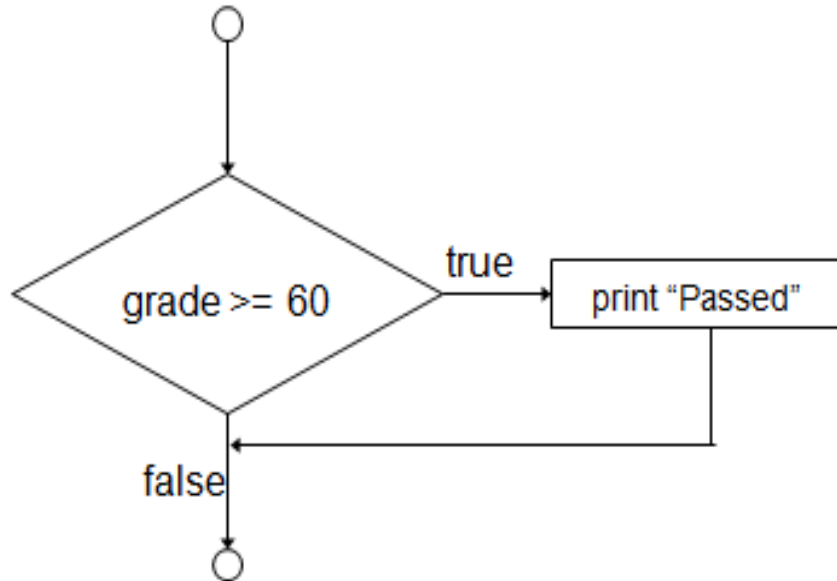
- Translation of pseudocode statement into C++:

```
if ( grade >= 60 )  
    cout << "Passed";
```

- Diamond symbol (decision symbol)
 - indicates decision is to be made
 - Contains an expression that can be true or false.
 - Test the condition, follow appropriate path
- **if** structure is a single-entry/single-exit structure

5. The if Selection Structure

- Flowchart of pseudocode statement



A decision can be made on any expression.

zero - **false**

nonzero - **true**

Example:

3 - 4 is true

5. The if Selection Structure

What is the Output?

```
int x = 5;
int y = 10;

if (x < y)
    ++x;
    ++y;

cout << " x = " << x << " y = " << y
      << endl;
```

5. The if Selection Structure

Example: Guess a secret number

```
# include <iostream.h>
# define secret 10
void main ( )
{
int n;
cout << " Enter a number: " ;
cin >> n ;
if ( n == secret )
cout << " You guessed right. " ;
else
cout<< " Try again, your guess is wrong." ;
}
```

5. The if Selection Structure

Example: Print a number squared if it is less than 100

```
# include <iostream.h>
void main ( )
{
int n, squared;
cout << " Enter a number: " << endl ;
cin >> n ;
if ( n < 100 ) {
squared = n * n ;
cout << squared ; }
else
cout<< " Your number is greater than 100" ;
}
```

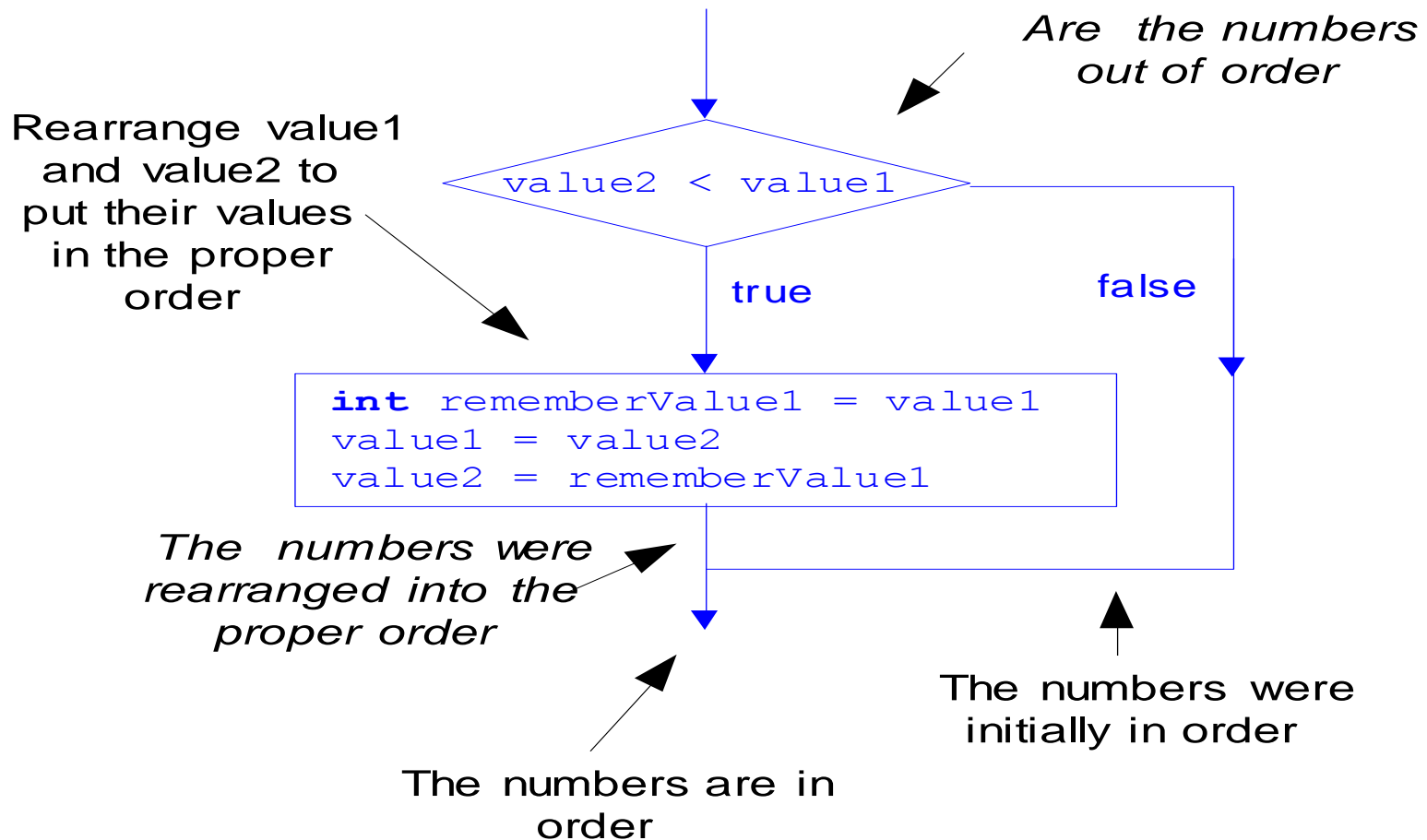
5. The if Selection Structure

Example: Sorting Two Numbers

```
cout << "Enter two integers: ";  
int Value1;  
int Value2;  
cin >> Value1 >> Value2;  
if (Value1 > Value2) {  
    int RememberValue1 = Value1;  
    Value1 = Value2;  
    Value2 = RememberValue1;  
}  
cout << "The input in sorted order: "  
    << Value1 << " " << Value2 << endl;
```


5. The if Selection Structure

Example: Sorting Two Numbers - Semantics



6. The `if/else` Selection Structure

- **`if`**
 - Only performs an action if the condition is true
- **`if/else`**
 - A different action is performed when condition is true and when condition is false

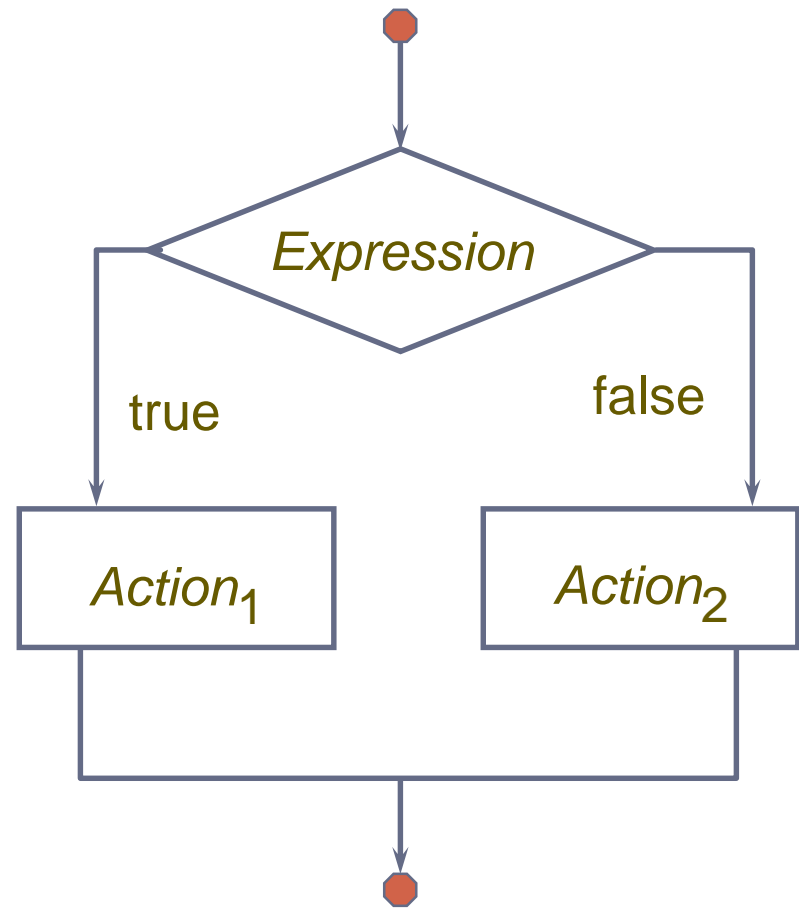
6. The `if/else` Selection Structure

- Syntax

```
if (Expression)  
    Action1  
else  
    Action2
```

- If *Expression* is true then execute *Action*₁ otherwise execute *Action*₂

```
if (v == 0) {  
    cout << "v is 0";  
}  
else {  
    cout << "v is not 0";  
}
```



6. The `if/else` Selection Structure

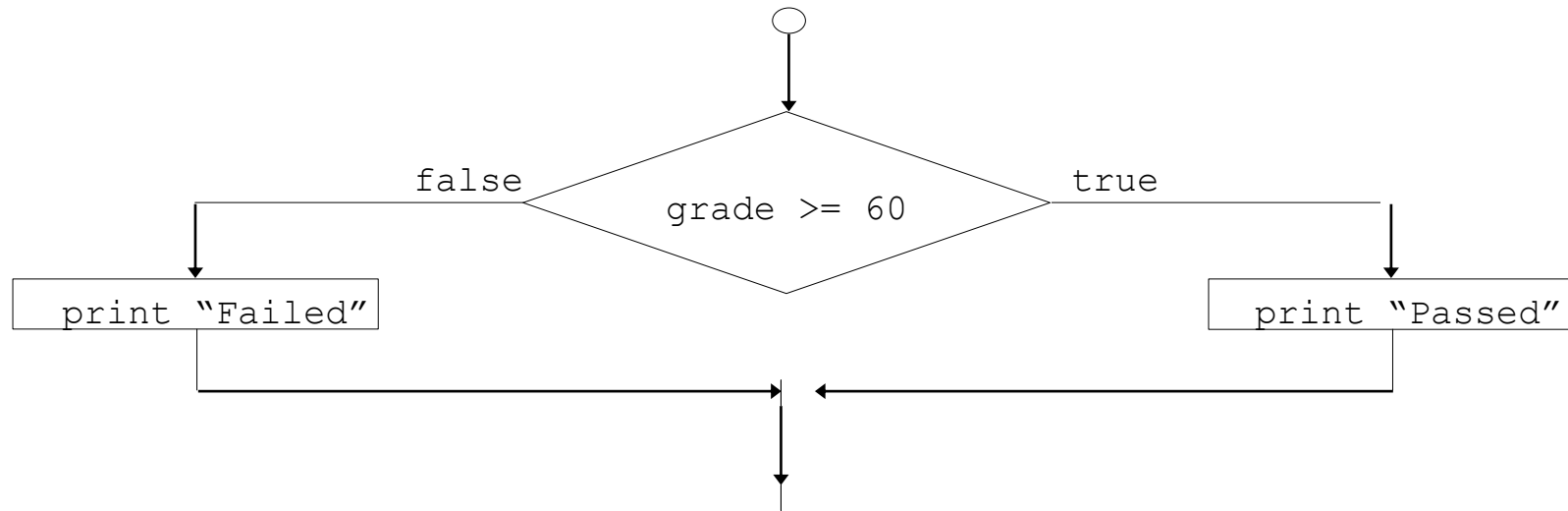
- Psuedocode

```
if student's grade is greater than or equal to 60  
    print "Passed"  
else  
    print "Failed"
```

- C++ code

```
if ( grade >= 60 )  
    cout << "Passed";  
else  
    cout << "Failed";
```

6. The `if/else` Selection Structure



- Ternary conditional operator (`? :`)
 - Takes three arguments (condition, value if **true**, value if **false**)
- Our pseudocode could be written:

```
cout << ( grade >= 60 ? "Passed" :  
  "Failed" );
```

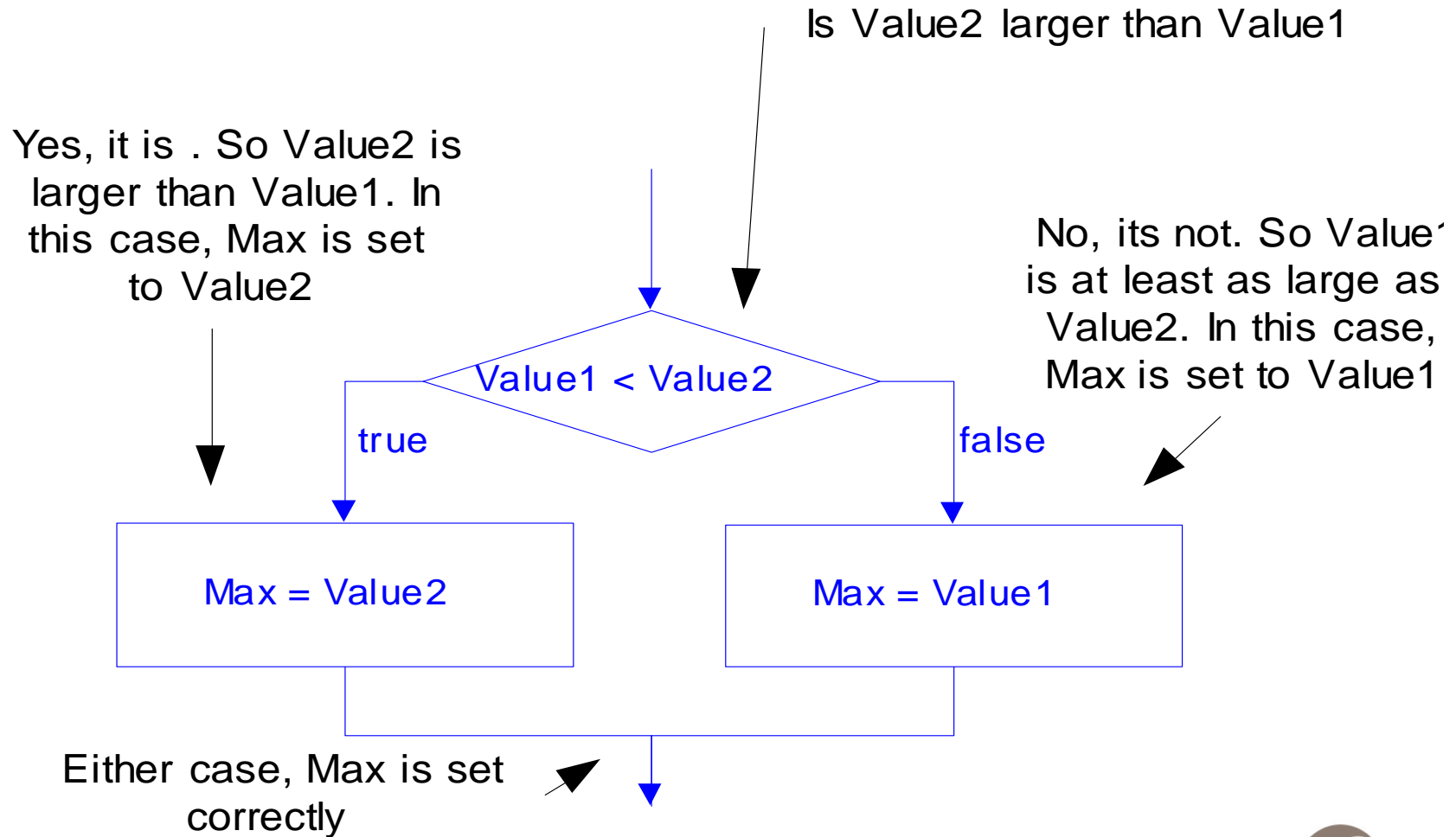
6. The `if/else` Selection Structure

Example: Finding the Maximum

```
cout << "Enter two integers: ";
int Value1;
int Value2;
cin >> Value1 >> Value2;
int Max;
if (Value1 < Value2) {
    Max = Value2;
}
else {
    Max = Value1;
}
cout << "Maximum of inputs is: " << Max <<
endl;
```

6. The `if/else` Selection Structure

Example: Finding the Maximum - Semantics



6. The `if/else` Selection Structure

- Nested `if/else` structures

- Test for multiple cases by placing `if/else` selection structures inside `if/else` selection structures.

if student's grade is greater than or equal to 90

Print "A"

else

if student's grade is greater than or equal to 80

Print "B"

else

if student's grade is greater than or equal to 70

Print "C"

else

if student's grade is greater than or equal to 60

Print "D"

else

Print "F"

- Once a condition is met, the rest of the statements are skipped



6. The `if/else` Selection Structure

- Compound statement:

- Set of statements within a pair of braces

- Example:

```
if ( grade >= 60 )
    cout << "Passed.\n";
else {
    cout << "Failed.\n";
    cout << "You must take this course
again.\n";
}
```

- Without the braces,

```
cout << "You must take this course again.\n";
```

would be automatically executed

- Block

- Compound statements with declarations

6. The `if/else` Selection Structure

Example: Convert a student degree to a grade

```
#include <iostream.h>
void main ( )
{
    int degree ;
    cout << " Please enter your degree, it should be in the range from 0 to 100 " ;
    cin >> degree ;
    if ( degree >= 0 && degree <= 100 ) {
        if ( degree >= 90)
            cout << " Excellent....Your grade is A " ;
        else if ( degree >= 80)
            cout << " Very Good....Your grade is B " ;
        else if ( degree >= 70 )
            cout << " Good...Your grade is C " ;
        else
            cout << " You Fail " ;
    }
    else
        cout << " You enter a wrong degree, you should enter a number between 0 and 100";
}
```

| Grade | Condition |
|-------|---------------------------------------|
| A | Degree is greater than or equal to 90 |
| B | Degree is greater than or equal to 80 |
| C | Degree is greater than or equal to 70 |
| Fail | Degree is less than 70 |

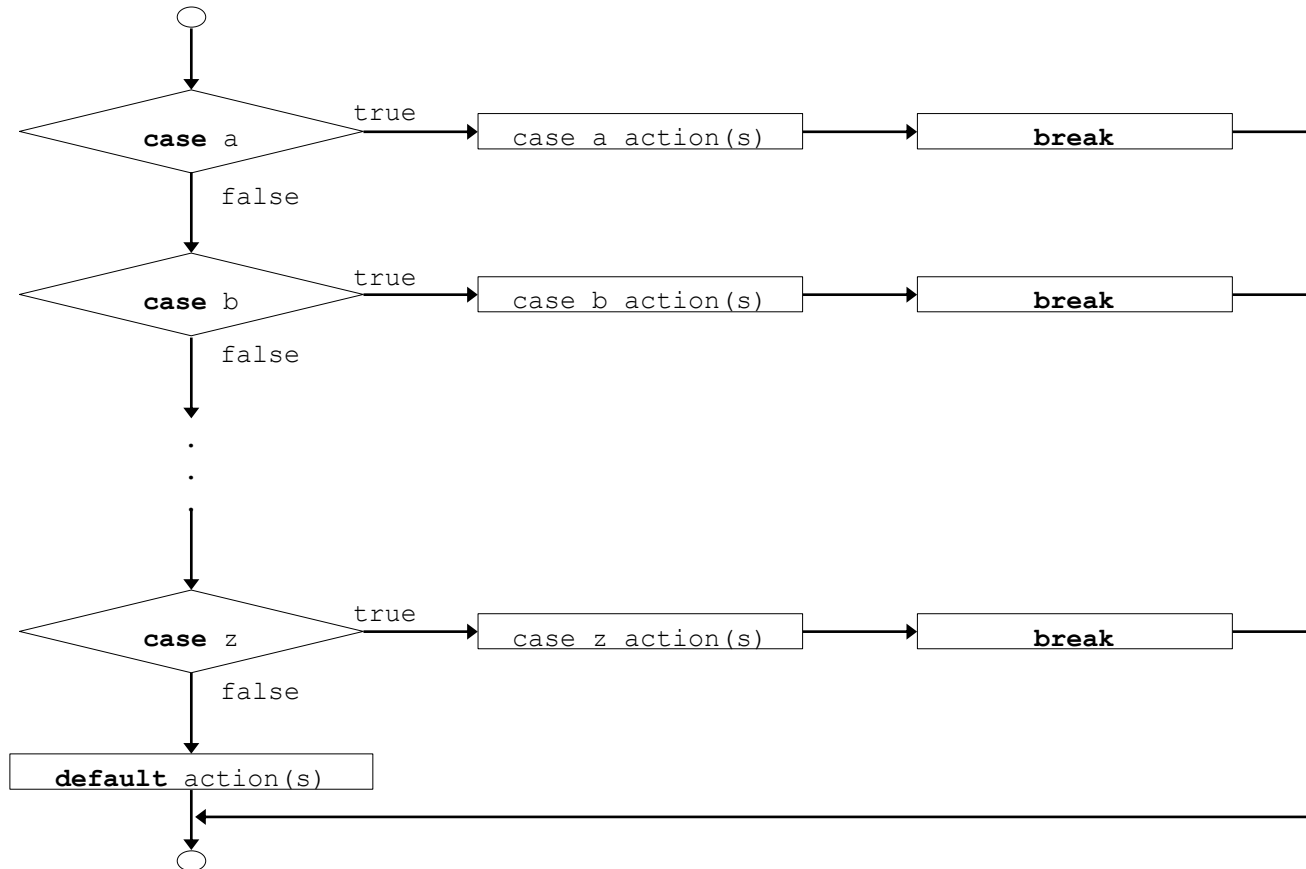
6. The `if/else` Selection Structure

- Syntax errors
 - Errors caught by compiler
- Logic errors
 - Errors which have their effect at execution time
 - Non-fatal logic errors
 - program runs, but has incorrect output
 - Fatal logic errors
 - program exits prematurely

7. The Switch Multiple-Selection Structure

- **switch**

- Useful when variable or expression is tested for multiple values
- Consists of a series of **case** labels and an optional **default** case



7. The Switch Multiple-Selection Structure

Example: Decide a letter is vowel or not

```
switch (ch) {
    case 'a': case 'A':
    case 'e': case 'E':
    case 'i': case 'I':
    case 'o': case 'O':
    case 'u': case 'U':
        cout << ch << " is a vowel" <<
endl;
        break;
    default:
        cout << ch << " is not a vowel" <<
endl;
}
```



Outline



1. Initialize variables

2. Input data

2.1 Use switch loop to update count

```
1 // Fig. 2.22: fig02 22.cpp
2 // Counting letter grades
3 #include <iostream>
4
5 using std::cout;
6 using std::cin;
7 using std::endl;
8
9 int main()
10 {
11     int grade,          // one grade
12         aCount = 0,    // number of A's
13         bCount = 0,    // number of B's
14         cCount = 0,    // number of C's
15         dCount = 0,    // number of D's
16         fCount = 0;    // number of F's
17
18     cout << "Enter the letter grades." << endl
19          << "Enter the EOF character to end input." << endl;
20
21     while ( ( grade = cin.get() ) != EOF ) {
22
23         switch ( grade ) {
24
25             case 'A': // grade was uppercase A
26                 case 'a': // or lowercase a
27                     ++aCount;
28                     break; // necessary to exit switch
29
30             case 'B': // grade was uppercase B
31                 case 'b': // or lowercase b
32                     ++bCount;
33                     break;
34
```

Notice how the **case** statement is used



2.1 Use switch loop to update count

3. Print results

break causes switch to end and the program continues with the first statement after the switch structure.

Notice the default statement.

```
35 case 'C': // grade was uppercase C
36 case 'c': // or lowercase c
37     ++cCount;
38     break;
39
40 case 'D': // grade was uppercase
41 case 'd': // or lowercase d
42     ++dCount;
43     break;
44
45 case 'F': // grade was uppercase
46 case 'f': // or lowercase f
47     ++fCount;
48     break;
49
50 case '\n': // ignore newlines,
51 case '\t': // tabs,
52 case ' ': // and spaces in input
53     break;
54
55 default: // catch all other characters
56     cout << "Incorrect letter grade entered."
57         << " Enter a new grade." << endl;
58     break; // optional
59 }
60 }
61
62 cout << "\n\nTotals for each letter grade are:"
63     << "\nA: " << aCount
64     << "\nB: " << bCount
65     << "\nC: " << cCount
66     << "\nD: " << dCount
67     << "\nF: " << fCount << endl;
68
69 return 0;
70 }
```



Outline

Program Output

Enter the letter grades.

Enter the EOF character to end input.

a

B

c

C

A

d

f

C

E

Incorrect letter grade entered. Enter a new grade.

D

A

b

Totals for each letter grade are:

A: 3

B: 2

C: 3

D: 2

F: 1

7. The Switch Multiple-Selection Structure

Example: Convert a student degree to a grade

```
#include <iostream.h>
void main ( )
{
    int degree, temp ;
    cout << " Please enter your degree, it should be in the range from 0 to 100 " ;
    cin >> degree ;
    temp = degree / 10;
    switch ( temp) {
        case 10 :
        case 9 :
            cout << " Excellent....Your grade is A " ;
            break ;
        case 8 :
            cout << " Very Good....Your grade is B " ;
            break ;
        case 7 :
            cout << " Good...Your grade is C " ;
            break ;
        case 6 : case 5: case 4: case 3: case 2: case 1: case 0:
            cout << " You Fail " ;
            break;
        default :
            cout << " You enter a wrong degree, you should enter a number between 0 and 100";
    }
}
```

| Grade | Condition |
|-------|---------------------------------------|
| A | Degree is greater than or equal to 90 |
| B | Degree is greater than or equal to 80 |
| C | Degree is greater than or equal to 70 |
| Fail | Degree is less than 70 |

7. The Switch Multiple-Selection Structure

Example: Determine the number of days in a month

```
# include <iostream.h>
void main ( )
{
int month, year ;
cout << " Please enter the number of the month " ;
cin >> month ;
switch ( month ) {
case 1: case 3: case 5: case 7: case 8: case 10: case 12:
cout << " The number of days in this month is 31 days " << endl ;
break ;
case 4: case 6: case 9: case 11:
cout << " The number of days in this month is 30 days " << endl ;
break ;
case 2:
cout << " Please enter the year:" ;
cin >> year ;
if ( year % 400 == 0 )
cout << " The number of days in this month is 29 days " << endl ;
else
cout << " The number of days in this month is 28 days " << endl ;
break ;
default :
cout << " The month number should be in the range from 1 to 12 " ;
} }
}
```