جامعة الملك سعود
King Saud University

Control Structures
Decisive making in C++

Lecture 3

Description

In this chapter we will learn the conditional statement such as if statement and switch case which is used to control different action for different decision.

Control Structures

- Sequential execution
  - Statements executed in order
- Transfer of control
  - Next statement executed *not* next one in sequence
- 3 control structures (Bohm and Jacopini)
  - Sequence structure
    - Programs executed sequentially by default
  - Selection structures
    - **`if`**, **`if/else`**, **`switch`**
  - Repetition structures
    - **`while`**, **`do/while`**, **`for`**

Control Structures

- C++ keywords
  - Cannot be used as identifiers or variable names

C++ Keywords				
Keywords common to the C and C++ programming languages				
auto	break	case	char	const
continue	default	do	double	else
enum	extern	float	for	goto
if	int	long	register	return
short	signed	sizeof	static	struct
switch	typedef	union	unsigned	void
volatile	while			
C++ only keywords				
asm	bool	catch	class	const_cast
delete	dynamic_cast	explicit	false	friend
inline	mutable	namespace	new	operator
private	protected	public	reinterpret_cast	
static_cast	template	this	throw	true
try	typeid	typename	using	virtual
wchar_t				

Control Structures

- Flowchart
  - Graphical representation of an algorithm
  - Special-purpose symbols connected by arrows (flowlines)
  - Rectangle symbol (action symbol)
    - Any type of action
  - Oval symbol
    - Beginning or end of a program, or a section of code (circles)

Control Structures

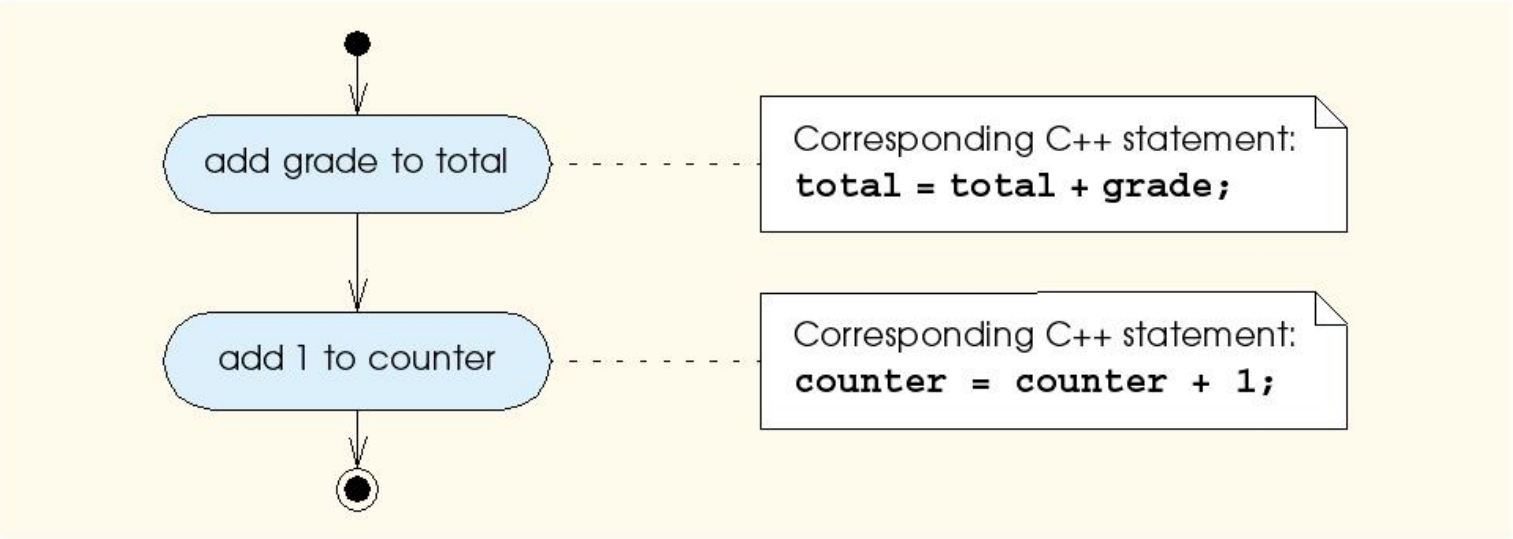


Fig. 2.1 Sequence structure activity diagram.

if Selection Structure

- Selection structure
  - Choose among alternative courses of action
  - Pseudocode example:

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

    Print "Passed"
  - If the condition is **`true`**
    - Print statement executed, program continues to next statement
  - If the condition is **`false`**
    - Print statement ignored, program continues
  - Indenting makes programs easier to read
    - C++ ignores whitespace characters (tabs, spaces, etc.)

if Selection Structure

- Translation into C++

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

  Print "Passed"

```
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 condition, follow path
- **`if`** structure
  - Single-entry/single-exit

if Selection Structure

- Flowchart of pseudocode statement

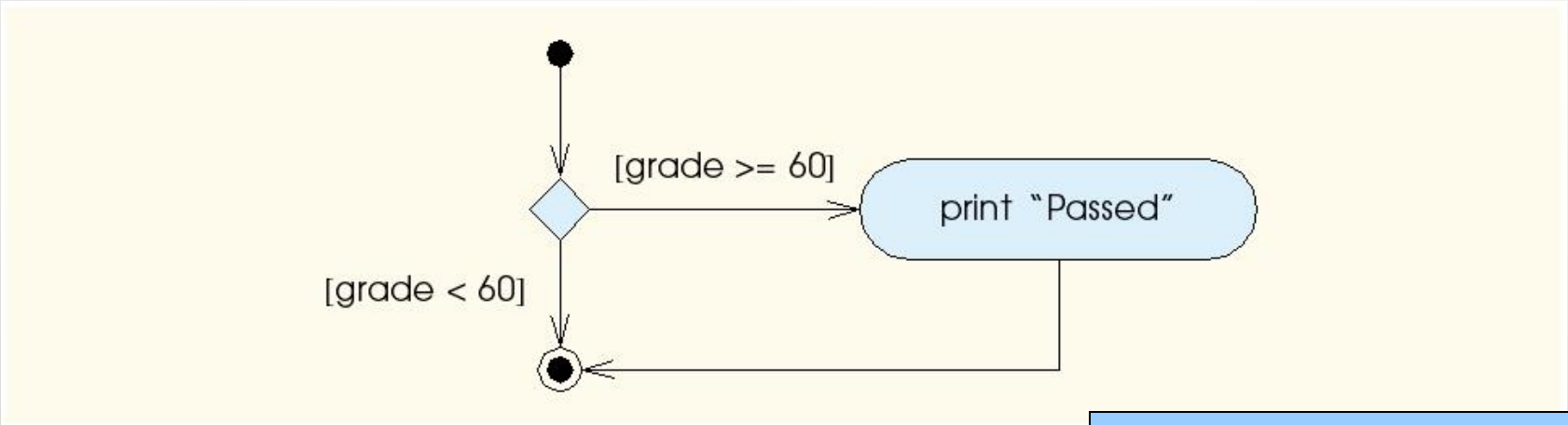


Fig. 2.3 **if** single-selection structure activity diagram.

A decision can be made on any expression.

zero - **`false`**

nonzero - **`true`**

Example:

`3 - 4` is **`true`**

`if/else` Selection Structure

- **`if`**
  - Performs action if condition true
- **`if/else`**
  - Different actions if conditions true or false
- Pseudocode

  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";
```

`if/else` Selection Structure

- Ternary conditional operator (**`?:`**)
  - Three arguments (condition, value if **`true`**, value if **`false`**)
- Code could be written:

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

Condition **Value if true** **Value if false**

`if/else` Selection Structure

- Nested **`if/else`** structures
  - One inside another, test for multiple cases
  - Once condition met, other statements skipped

```
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"
```

`if/else if` Selection Structure

- Example

```
if ( grade >= 90 )        // 90 and above
   cout << "A";
else if ( grade >= 80 )   // 80-89
   cout << "B";
else if ( grade >= 70 )   // 70-79
   cout << "C";
else if ( grade >= 60 )   // 60-69
   cout << "D";
else                      // less than 60
   cout << "F";
```

`if/else` Selection Structure

- Compound statement
  - Set of statements within a pair of braces

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

  - Without braces,

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

always executed

- Block
  - Set of statements within braces

switch Multiple-Selection Structure

Test variable for multiple values

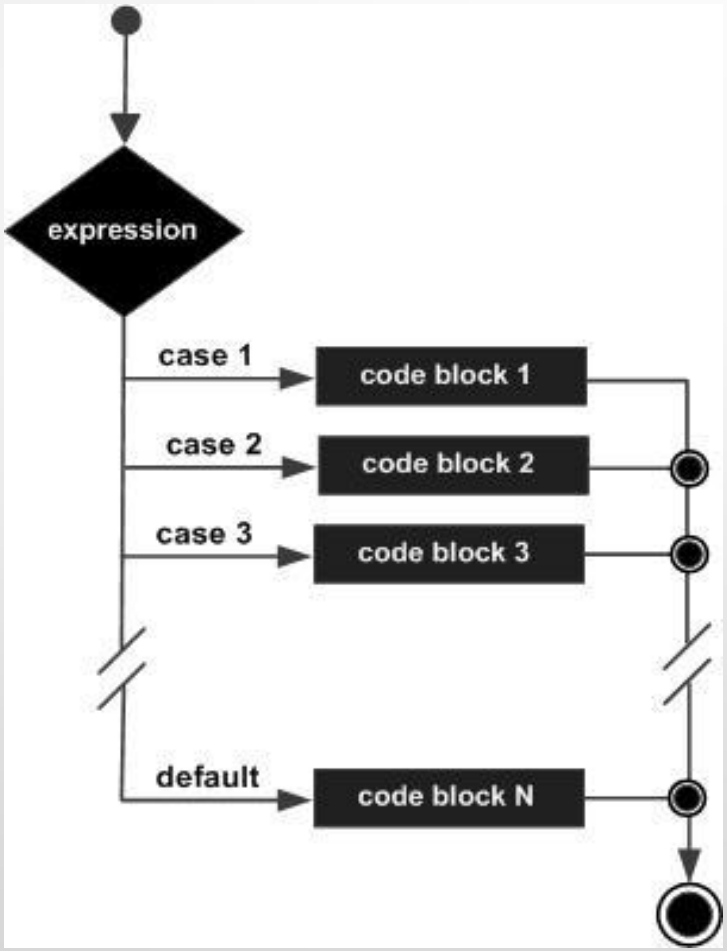
- Series of **`case`** labels and optional **`default`** case

```
switch ( variable ) {
 case value1:          // taken if variable == value1
 statements
 break;               // necessary to exit switch

 case value2:
 case value3:       // taken if variable == value2 or == value3
 statements
 break;

 default:         // taken if variable matches no other cases
 statements
  break;
}
```

- The following rules apply to a **switch** statement –
- The variable used in a switch statement can only be integers, convertable integers (byte, short, char), strings and enums.
- You can have any number of case statements within a switch. Each case is followed by the value to be compared to and a colon.
- The value for a case must be the same data type as the variable in the switch and it must be a constant or a literal.
- When the variable being switched on is equal to a case, the statements following that case will execute until a *break* statement is reached.

- When a break statement is reached, the switch terminates, and the flow of control jumps to the next line following the switch statement.
- Not every case needs to contain a break. If no break appears, the flow of control will *fall through* to subsequent cases until a break is reached.
- A *switch* statement can have an optional default case, which must appear at the end of the switch. The default case can be used for performing a task when none of the cases is true. No break is needed in the default case.

Flow Chart


expression
case 1
code block 1
case 2
code block 2
case 3
code block 3
default
code block N

```
public class Test {

  public static void main(String args[]) {
    // char grade = args[o].charAt(o);
    char grade = 'C';

    switch(grade) {
      case 'A' :
        System.out.println("Excellent!");
        break;
 case 'B' :
      case 'C' :
        System.out.println("Well done");
```

```
 break;
case 'D' :
        System.out.println("You passed");
      case 'F' :
        System.out.println("Better try again");
        break;
      default :
        System.out.println("Invalid grade");
    }
    System.out.println("Your grade is " + grade);
  }
}
```

Summary

- If condition statement
- Flow chart for if condition statement
- Compound statement
- If-else statement
- If-else if statement
- Switch statement and its flow chart

Thank You