

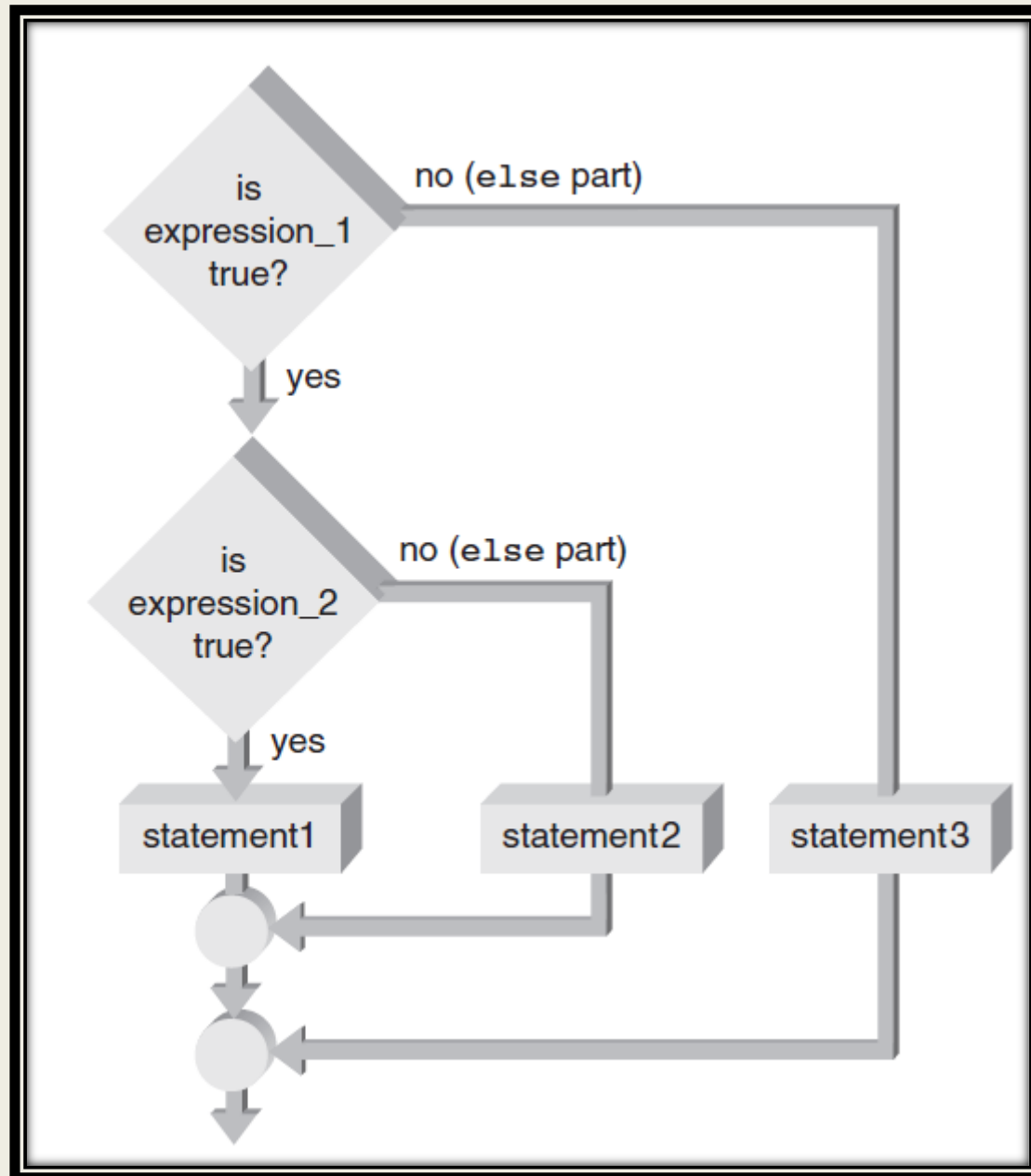
# **PROGRAMMING FUNDAMENTALS LECTURE 6**

Ahmed kamil

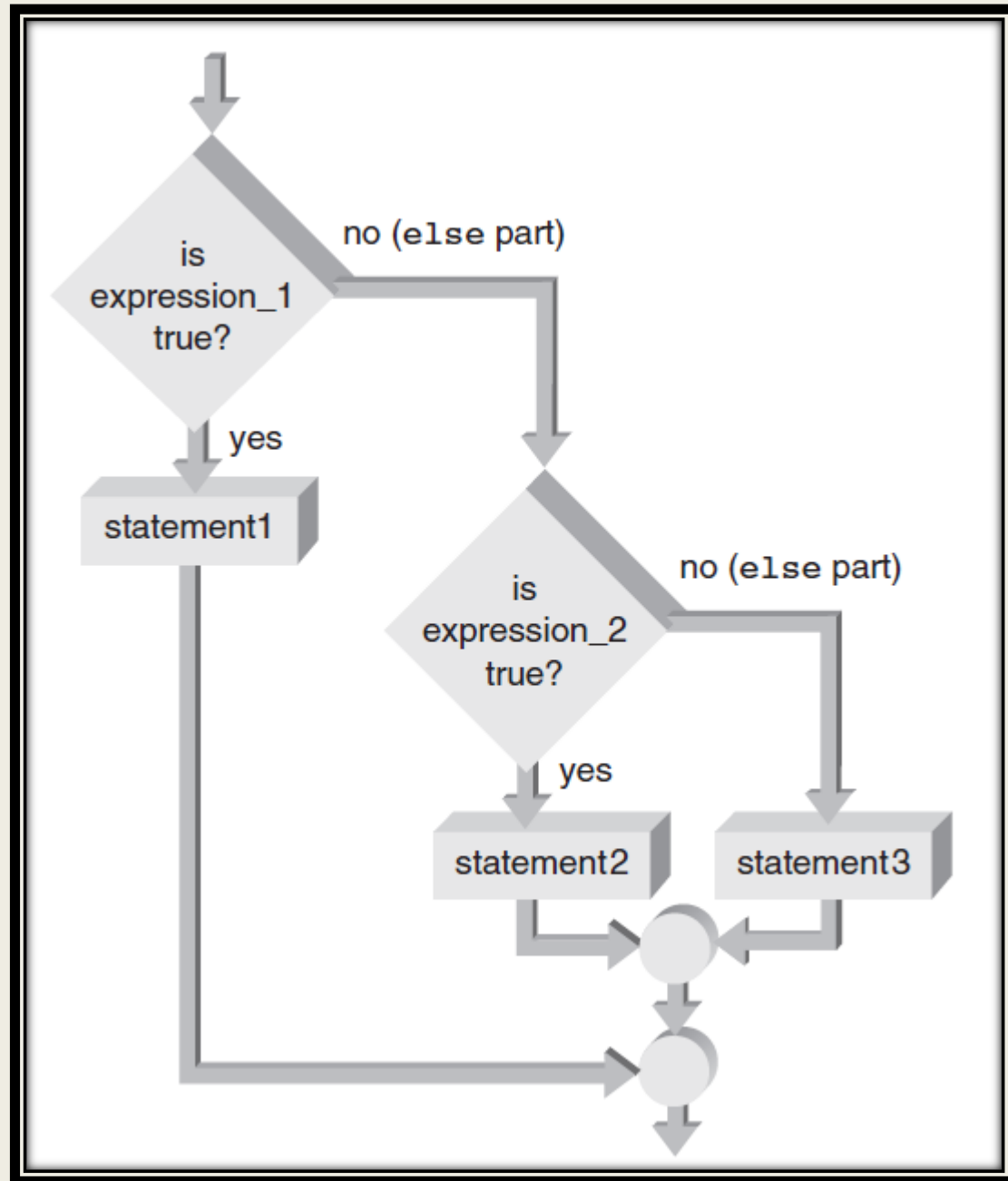
# Lecture Objectives

- Learn what is “nested if statement”.
- Learn what is “if – else chain”.
- Learn what is “switch statements”.

# Syntax: Nested within the if part



# Syntax: Nested within the else part



# Example

- write program to read the user gender and his age. the program prints "you are welcome in our club ", if the gender is M and age is greater than 21 otherwise it's prints "sorry, you are out of our conditions".

# if-else Chain

```
if (expression_1)
statement1;
else if (expression_2)
statement2;
else if (expression_3)
statement3;
.
.
.
else if (expression_n)
statementn;
else
last_statement;
```

# Example:

The grade level of undergraduate college students is typically determined according to the following schedule:

Number of Credits Completed	Grade Level
Less than 32	Freshman
32 to 63	Sophomore
64 to 95	Junior
96 or more	Senior

# Example:

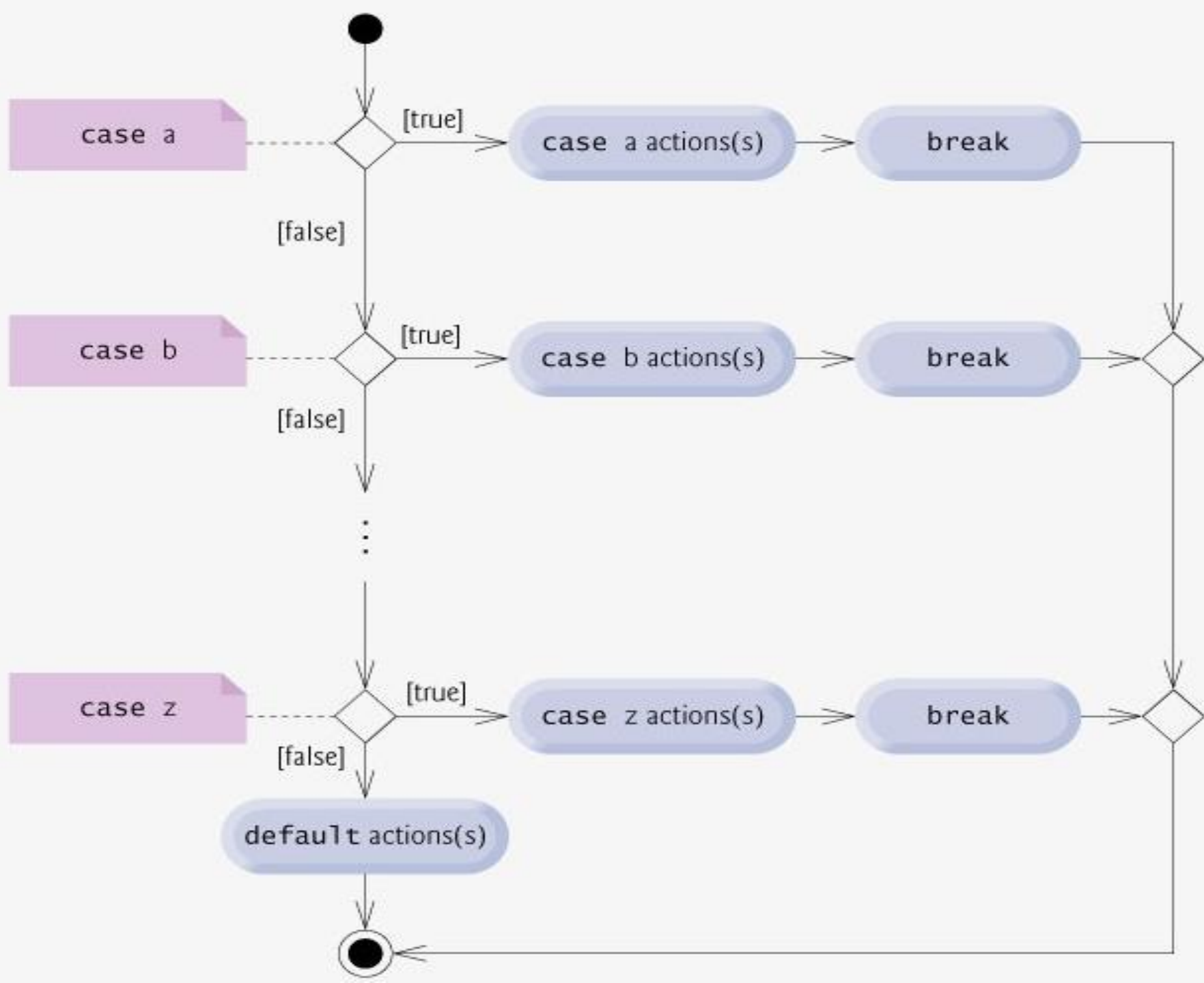
A student's letter grade is calculated according to the following schedule:

Numerical Grade	Letter Grade
Greater than or equal to 90	A
Less than 90 but greater than or equal to 80	B
Less than 80 but greater than or equal to 70	C
Less than 70 but greater than or equal to 60	D
Less than 60	F



# The **switch** Statement

- Used to select among statements from several alternatives
- May sometimes be used instead of **if/else if** statements



# switch Statement Format

```
switch (IntExpression)
{
    case exp1: statement set 1;
    case exp2: statement set 2;
    ...
    case expn: statement set n;
    default:   statement set n+1;
}
```

# Example

```
cout << "Enter simple expression: ";  
int Left;  
int Right;  
char Operator;  
cin >> Left >> Operator >> Right;  
cout << Left << " " << Operator << " " << Right  
    << " = ";  
switch (Operator) {  
    case '+' : cout << Left + Right << endl; break;  
    case '-' : cout << Left - Right << endl; break;  
    case '*' : cout << Left * Right << endl; break;  
    case '/' : cout << Left / Right << endl; break;  
    default: cout << "Illegal operation" << endl;  
}
```

# **switch** Statement Requirements

- 1) ***IntExpression*** must be a **char** or an integer variable or an expression that evaluates to an integer value
- 2) ***exp1*** through ***expn*** must be constant integer type expressions and must be unique in the **switch** statement
- 3) **default** is optional but recommended

# How the **switch** Statement Works

- 1) *IntExpression* is evaluated
- 2) The value of *intExpression* is compared against *exp1* through *expn*.
- 3) If *IntExpression* matches value *expi*, the program branches to the statement(s) following *expi* and continues to the end of the **switch**
- 4) If no matching value is found, the program branches to the statement after **default**:

# The **break** Statement

- Used to stop execution in the current block
- Also used to exit a **switch** statement
- Useful to execute a single **case** statement without executing statements following it

# Example

```
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;  
}
```



# Example

```
switch (Letter)
{
    case 'a':
    case 'A': cout << "You Entered A.\n";
               break;
    case 'b':
    case 'B': cout << " You Entered B.\n";
               break;
    case 'c':
    case 'C': cout << " You Entered C.\n";
               break;
    default:  cout << "That is an invalid choice.\n";
}
```

# Exercise

Convert the following segment of code to switch statement:

```
int j, n;  
.....  
If (j==3 || j==5)  
    n = 6;  
else if ( j==4 || j==8)  
    n = 9;  
else if (j==2)  
    n = 8;  
else  
    n=0;
```

Write a program to read a number and convert this number to the equivalent word.

# Style ideas for the switch statement

- Unless you have many conditions (4 or more), use **if-else-if** instead of **switch**
- Always provide a **default** case – if you are pretty sure you have all cases covered, putting an error message in the default is good to identify unexpected errors
- Order the cases in some logical order (numeric, alphabetic)
- Keep the size of each of the **cases** small (If you have to do lots of work in each case, call a function from inside the case)

# Notes

- Without a **break** statement, execution will fall through to the next **case** label
- The expression of a `switch` statement must result in an ***integral type***, meaning an integer (`byte`, `short`, `int`, `long`) or a `char`
- It **cannot** be a `boolean` value or a floating point value (`float` or `double`)
- The implicit boolean condition in a `switch` statement is **equality**
- You cannot perform relational checks with a **`switch`** statement

Thank You for  
Listening.