M’Hamed BOUGARA university

Institute of Electrical and Electronic Engineering (IGEE)

EE 121. Office suite

Lecture 3: Introduction to programming and C programming language

Dr. F. Kerouh

Electronics Department

IGEE, UMBB

Part III. Program looping

III.1 Conditional

- if (expr) { … } else {…}
- switch (expr) { case c1: case c2: … }

III.2 Iteration

- while (expr) { … } zero or more iterations
- do … while (expr) at least one iteration
- for (init ; valid ; next) { … }

III.3 Jump

- goto label

Part III.1 Decision making elements

Outlines

III.1. 1 Introduction to Decision Making Statements

III.1. 2 If statement

- ✓ Examples

III.1. 3 If-else statement

- ✓ Examples

III.1. 4 Switch statement

- ✓ Examples

III.1.1 Introduction

- ✓ Decision making elements are used to have a program that executes different statements depending on certain conditions.

- ✓ In a sense, they make a program "smarter" by allowing different choices to be made. In C, there are three decision making statements:

  1. **if execute a statement or not**
  2. **if-else choose to execute one of two statements**
  3. **switch choose to execute one of a number of statements**

III.1.2 If statement

- ✓ The if statement allows branching (decision making) depending upon a condition. Program code is executed or skipped.

The basic syntax is

if (*control expression)*

program statement;

- ✓ If the control expression is TRUE, the body of the if statement is executed. If it is FALSE, the body of the if statement is skipped.

III.1.2.1 If statement examples

- ✓ **Example 1:**

```
if (x!=0)
  y/=x;
```

- ✓ **Example 2:**

```
 if (grade>=90)
 {
printf("\nYour grade is %d",grade);
printf("\nCongratulations!");
 }
```

- ✓ **Example 3:**

```
if (nbr1>=nbr2)
if (nbr1>=nbr3)
printf(“%d is the greater number“,nbr1);
```

```
      if (( nbr1>=nbr2 ) && ( nbr1>=nbr3) )
printf(“%d is the greater number”,nbr1);
```

III.1.2. 2 If –else statement

Used to decide between two courses of action. The syntax of the if-else statement is:

```
if (expression)
  statement1;
else
  statement2;
```

- If the expression is TRUE, *statement1 is executed; statement2 is* skipped.
- If the expression is FALSE, *statement2 is executed; statement1 is* skipped.

III.1.2.2 If –else statement: Examples

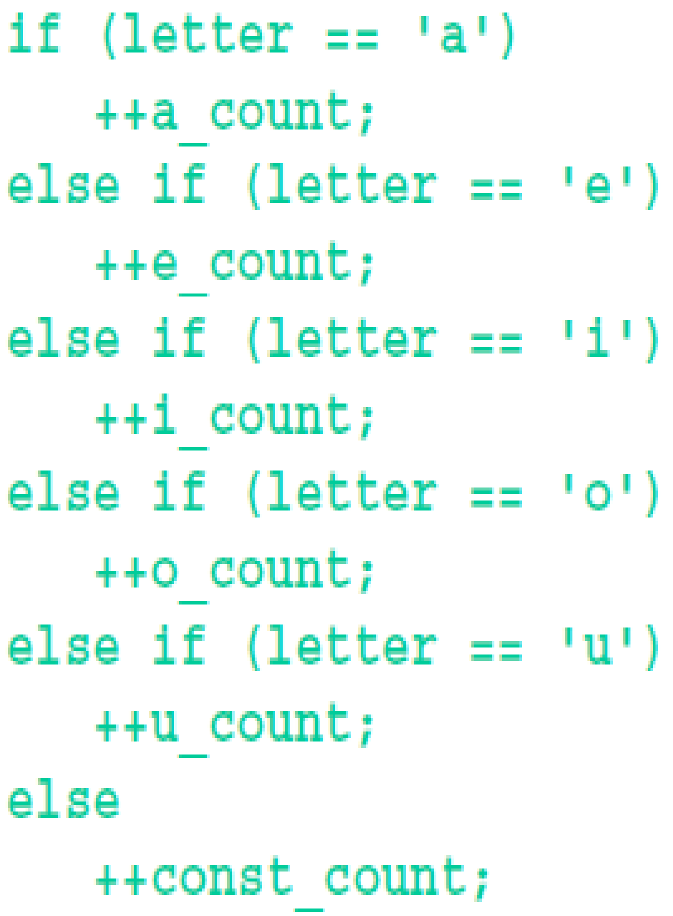
```
if (x<y)
    min=x;
else
    min=y;
```

```
if (letter == 'e') {
    ++e_count;
    ++vowel_count; }
else
    ++other_count;
```

III.1.2. 3 If –else Ladder

- ✓Write a program that counts how many vowels there are in a piece of text?

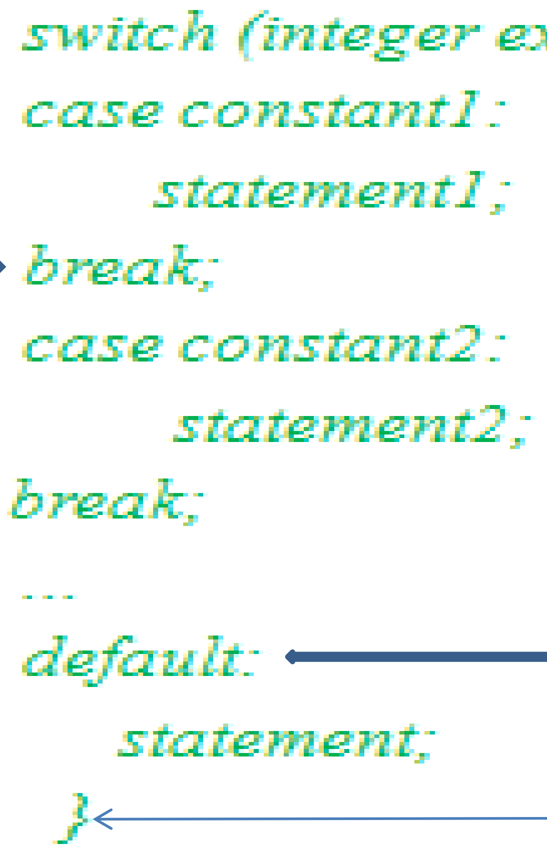
- ✓This is possible by nesting **if-else statements together to** make what is called an **if-else ladder.**



```
if (letter == 'a')
   ++a_count;
else if (letter == 'e')
   ++e_count;
else if (letter == 'i')
   ++i_count;
else if (letter == 'o')
   ++o_count;
else if (letter == 'u')
   ++u_count;
else
   ++const_count;
```

III.1. 3 switch statement

- ✓ It is a better way of writing a program which employs an if-else ladder. It is C’s built-in multiple branch decision statement. The syntax for the switch statement is as follows:



```
switch (integer expression) {
case constant1 :
      statement1 ;
break;
case constant2 :
       statement2 ;
break;

...
default:
     statement;
  }
```

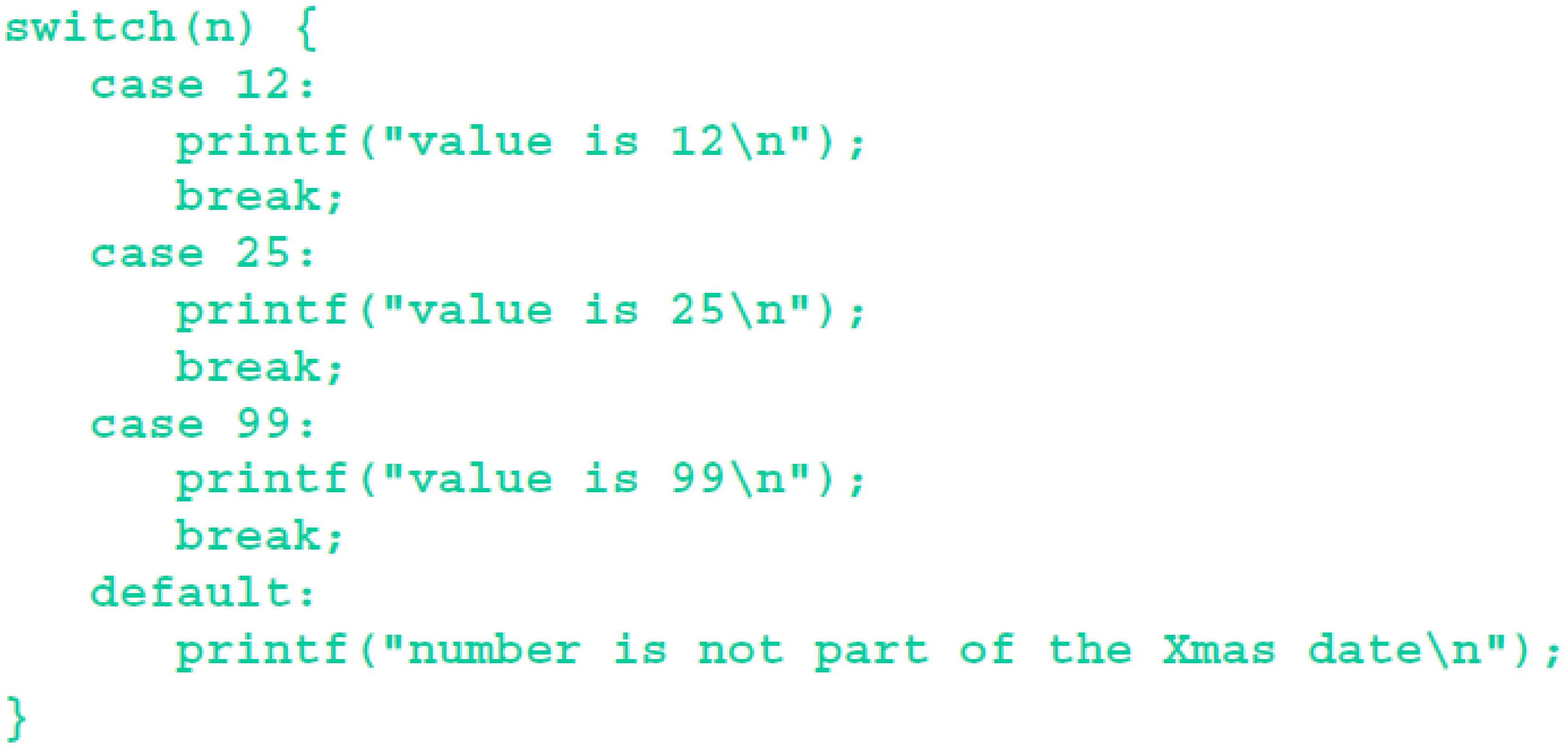
✓Should be included at the end of each case statement.

✓It causes an exit from the switch shunt.

Marks the Begin and the end of the switch statement

Optional: equivalent to else used with if statement

III.1. 3 switch statement: Example



```
switch(n) {
   case 12:
      printf("value is 12\n");
      break;
   case 25:
      printf("value is 25\n");
      break;
   case 99:
      printf("value is 99\n");
      break;
   default:
      printf("number is not part of the Xmas date\n");
}
```

III.1. 3 switch statement

- ✓ The switch statement works as follows:

  1. Integer control expression is evaluated.

  2. A match is looked for between this expression value and the case *constants. If a match is found, execute the statements for that case. If a* match is not found, execute the default statement.

  3. Terminate switch when a break statement is encountered or by "falling out the end".

III.1. 3 switch statement

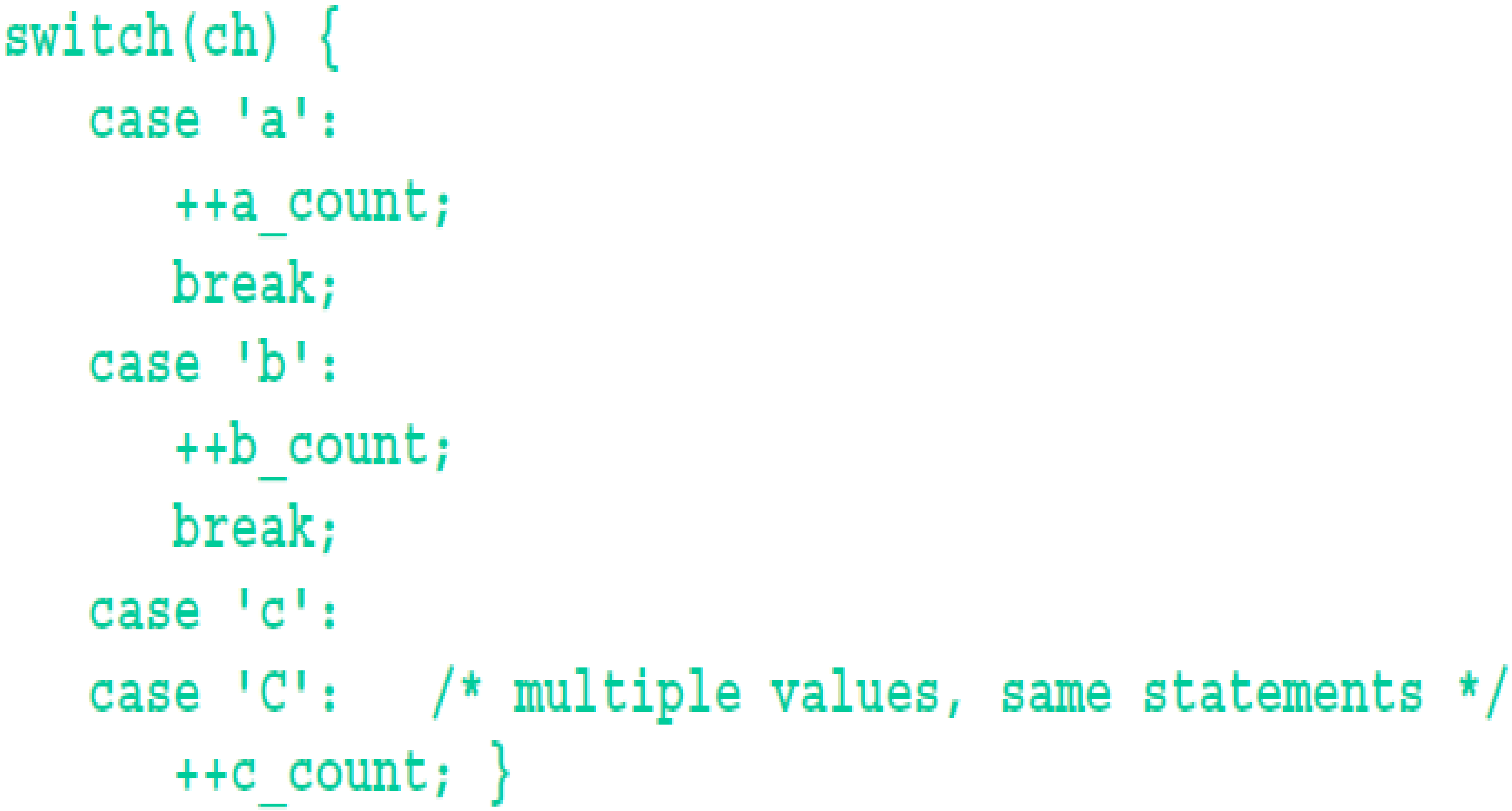
✓ Important rematks while using a switch statement:

1. Case values must be unique.

2. Switch statement only tests for equality.

3. The control expression can be of type character since they are internally treated as integers.

III.1. 3 switch statement: Example



```
switch(ch) {
   case 'a':
      ++a_count;
      break;
   case 'b':
      ++b_count;
      break;
   case 'c':
   case 'C':   /* multiple values, same statements */
      ++c_count; }
```

III.1. 4 Conclusion

Law of **Mosher** about software development:

“Don’t worry if it doesn’t work right. If everything did, you’d be out of a job.”…