

Capítulo 5

This activity contains 31 questions.

1.

Section 5.2 Essentials of Counter-Controlled Repetition

5.2 Q1: Which of the following does counter-controlled repetition require?

- ☐ An initial value.
- ☐ Counter-controlled repetition requires all of the above.
- ☐ An increment or decrement by which the control variable is modified each time through the loop.
- ☐ A condition that tests for the final value.

2.

5.2 Q2: The statement

```
while ( --counter >= 1 )  
counter % 2 ? cout << "A" : cout << "B";
```

cannot be rewritten as:

- ☐

```
while ( --counter >= 1 )  
    if ( counter % 2 )  
        cout << "A";  
    else  
        cout << "B";.
```
- ☐

```
while ( --counter >= 1 )  
    counter % 2 == 0 ? cout << "B" : cout << "A";.
```
- ☐

```
while ( counter >= 1 )  
    if ( counter % 2 )  
        cout << "A";  
    else  
        cout << "B";.  
--counter;
```
- ☐

```
while ( counter > 1 )  
{  
    --counter;
```

```
if ( counter % 2 )  
    cout << "A";  
else  
    cout << "B";  
}.
```

3.

5.2 Q3: Which of the following is a bad programming practice?

- ☐ Indenting the statements in the body of each control structure.
- ☐ Placing vertical spacing above and below control structures.
- ☐ Using floating-point values for counters in counter-controlled repetition.
- ☐ Nesting multiple repetition structures one within another.

4.

Section 5.3 for Repetition Statement

5.3 Q1: If a variable is declared in the initialization expression of a for structure, then:

- ☐ The scope of the variable is restricted to that particular for loop.
- ☐ It is automatically reinitialized to zero once the loop is finished.
- ☐ It retains its final value after the loop is finished.
- ☐ It can not be used in any structures that are nested in that for structure.

5.

5.3 Q2: Which of the following is not true?

- ☐ A for loop can always be used to replace a while loop, and vice versa.
- ☐ The initialization and increment expressions can be comma-separated lists.
- ☐ The three expressions in the for structure are optional.
- ☐ You must declare the control variable outside of the for loop.

6.

5.3 Q3: Consider the execution of the following for loop

```
for (int x = 1; x < 5; increment )  
    cout << x + 1 << endl;
```

If the last value printed is 5, which of the following might have been used for increment?

- ☐ `x += 1.`
- ☐ *Any of the above.*
- ☐ *a. x++.*
- ☐ `++x.`

7.

Section 5.4 Examples Using the for Statement

5.4 Q1: Which of the following for headers is not valid?

- ☐ `for (int i = 0: i < 10;).`
- ☐ `for (int i = 0; i < 10; i++).`
- ☐ `for (int i = 0; int j = 5; ; i++).`
- ☐ `int i = 0;`
`for (; i < 10; i++).`

8.

A Caution about Using Type double for Monetary Amounts

5.4 Q2: float and double variables should be used:

- ☐ *To perform monetary calculations.*
- ☐ *To store true/false values.*
- ☐ *As counters.*
- ☐ *As imprecise representations of decimal numbers.*

9.

Using Stream Manipulators to Format Numeric Output

5.4 Q3: Which of the following is a parameterized stream manipulator used to format output?

- ☐ `setw.`
- ☐ `left.`
- ☐ `right.`
- ☐ `fixed.`

10.

Section 5.5 do...while Repetition Statement

5.5 Q1: If a do...while structure is used:

- ☐ *Counter-controlled repetition is not possible.*
- ☐ *The body of the loop will execute at least once.*
- ☐ *An infinite loop will not take place.*

- ☐ An off-by-one error will not occur.

11.

5.5 Q2: What will the following program segment do?

```
int counter = 1;
do {
    cout << counter << " ";
} while ( ++counter <= 10 );
```

- ☐ Print the numbers 1 through 10.
- ☐ Cause a syntax error.
- ☐ Print the numbers 1 through 9.
- ☐ Print the numbers 1 through 11.

12.

Section 5.6 switch Multiple-Selection Statement

5.6 Q1: A switch statement should be used:

- ☐ To replace all if and if...else statements.
- ☐ As a multiple-selection structure.
- ☐ As a single-selection structure.
- ☐ As a double-selection structure.

13.

5.6 Q2: In a switch structure:

- ☐ Multiple actions do not need to be enclosed in braces.
- ☐ A break is required after each case.
- ☐ A break is required after the default case.
- ☐ A default case is required.

14.

5.6 Q3: Which of the following is correct when labeling cases in a switch structure?

- ☐ case1.
- ☐ Case 1.
- ☐ case 1.
- ☐ Case1.

15.

5.6 Q4: switch can be used to test:

- ☐ *string constants.*
- ☐ *float constants.*
- ☐ *int constants.*
- ☐ *all types of constants.*

16.

5.6 Q5: Which of the following data types can be used to represent integers?

- ☐ *All of the above.*
- ☐ *char.*
- ☐ *short.*
- ☐ *long.*

17.

*Section 5.7 break and continue Statements**5.7 Q1: Which of the following is false?*

- ☐ *continue statements skip the remaining statements in current iteration of the body of the loop in which they are embedded.*
- ☐ *continue and break statements may be embedded within all C++ structures.*
- ☐ *break statements exit from the loop in which they are embedded.*
- ☐ *break and continue statements alter the flow of control.*

18.

5.7 Q2: Which of the following is false?

- ☐ *You should always try to write the fastest, smallest code possible before attempting to make it simple and correct.*
- ☐ *Many programmers feel that break and continue violate structured programming.*
- ☐ *The effects of break and continue statements can be achieved by structured programming techniques.*
- ☐ *break and continue statements can perform faster than their corresponding structured techniques.*

19.

*Section 5.8 Logical Operators**5.8 Q1: In C++, the condition (4 > y > 1):*

- ☐ Does not evaluate correctly and should not be replaced by (4 > y && y > 1).
- ☐ Evaluates correctly and could not be replaced by (4 > y && y > 1).
- ☐ Does not evaluate correctly and should be replaced by (4 > y && y > 1).
- ☐ Evaluates correctly and could be replaced by (4 > y && y > 1).

20.

5.8 Q2: The OR (||) operator:

- ☐ Stops evaluation upon finding one condition to be true.
- ☐ Is a ternary operator.
- ☐ Associates from right to left.
- ☐ Has higher precedence than the AND (&&) operator.

21.

5.8 Q3: An operator that associates from right to left is:

- ☐ ..
- ☐ ().
- ☐ !=.
- ☐ ?:.

22.

5.8 Q4: The expression if (num != 65) cannot be replaced by:

- ☐ d. if (!(num - 65)).
- ☐ if (num > 65 || num < 65).
- ☐ if (!(num == 65)).
- ☐ if (num - 65).

23.

5.8 Q5: An example of a unary operator is:

- ☐ The % arithmetic operator.
- ☐ The = assignment operator.
- ☐ The ! logical operator.
- ☐ The < relational operator.

24.

Section 5.9 Confusing Equality (==) and Assignment (=) Operators

5.9 Q1: Variables are also known as:

- ☐ lvalues, and cannot be used as rvalues.
- ☐ Constant variables.
- ☐ lvalues, but can be used as rvalues.
- ☐ rvalues, and cannot be used as lvalues.

25.

5.9 Q2: Consider the following code, assuming that *x* is an *int* with an initial value of 12

```
if( x = 6 )  
    cout << x;
```

What is the output?

- ☐ 12.
- ☐ Nothing.
- ☐ 6.
- ☐ A syntax error is produced.

26.

5.9 Q3: Of the following, which is not a logic error?

- ☐ Using `==` to assign a value to a variable.
- ☐ Failing to initialize counter and total variables before the body of a loop.
- ☐ Not placing curly braces around the body of an *if* that contains two statements.
- ☐ Using commas instead of the two required semicolons in a *for* header.

27.

Section 5.10 Structured Programming Summary

5.10 Q1: The _____, _____ and _____ are the only three forms of control necessary.

- ☐ sequence, selection, repetition.
- ☐ break, continue, if...else.
- ☐ switch, if, else.
- ☐ for, while, do...while.

28.

5.10 Q2: Which of the following is not one of the C++ control structures?

- ☐ do...while.

- ☐ *switch.*
- ☐ *if.*
- ☐ *main.*

29.

5.10 Q3: Which of the following is not one the rules for forming structured programs?

- ☐ Any action state can be replaced by any control statement.
- ☐ Begin with the "simplest activity diagram."
- ☐ Any transition arrow can be reversed.
- ☐ Any action state can be replaced by two action states in sequence

30.

Section 5.11 (Optional) Software Engineering Case Study: Identifying Objects' States and Activities in the ATM System

5.11 Q1: Which of the following is not a part of a UML state diagram?

- ☐ Rounded rectangles representing states.
- ☐ Arrows with accompanying event description text representing transitions.
- ☐ A solid circle indicating the initial state.
- ☐ Fractions beside each state indicating the likelihood of entering that state.

31.

5.11 Q2: An activity diagram for modeling the actions involved in executing a balance inquiry transaction using the BalanceInquiry object should not include:

- ☐ Displaying the user's balance information on the screen.
- ☐ Retrieving the user's balance information from the database of accounts.
- ☐ All of the above actions should be modeled in this activity diagram.
- ☐ Receiving the user's main menu input indicating a desire to inquire the amount of his or her balance.

[Clear Answers / Start Over](#)[Submit Answers for Grading](#)

Answer choices in this exercise appear in a different order each time the page is loaded.



Copyright © 1995 - 2010 Pearson Education. All rights reserved.
[Legal Notice](#) | [Privacy Policy](#) | [Permissions](#)

