**Chapter 5: How to code control structures**

**Murach's C#**

**MULTIPLE CHOICE**

1. Which of the following operators is *not* a relational operator?

| a. | = | c. | > |
| --- | --- | --- | --- |
| b. | != | d. | <= |

2. Which of the following operators is *not* a logical operator?

| a. | ! | c. | > |
| --- | --- | --- | --- |
| b. | | | d. | && |

3. Which type of operator do you use to create a Boolean expression?

| a. | assignment | c. | logical |
| --- | --- | --- | --- |
| b. | relational | d. | conditional |

4. Which type of operator do you use to combine two or more Boolean expressions into a single expression?

| a. | assignment | c. | logical |
| --- | --- | --- | --- |
| b. | relational | d. | conditional |

5. What advantage does a conditional expression have over an if-else statement?

| a. | it’s easier to read |
| --- | --- |
| b. | it can use one or more else if clauses |
| c. | it requires less code |
| d. | it can have one or more statements for the if and else clauses |

6. If you use a short-circuit logical operator to combine two expressions

| a. | both expressions are always evaluated |
| --- | --- |
| b. | the second expression is evaluated only if it can affect the result |
| c. | the first expression is evaluated only if it can affect the result |
| d. | neither expression is evaluated |

7. Which of the following is a Boolean expression that tests if a decimal variable named salesThisYTD is equal to zero.

| a. | **salesThisYTD && 0** |
| --- | --- |
| b. | **salesThisYTD = 0** |
| c. | **salesThisYTD != 0** |
| d. | **salesThisYTD == 0** |

8. Which of the following Boolean expressions tests if a decimal variable named currentSales is greater than or equal to 1000 *or* a Boolean variable named newCustomer is equal to true?

| a. | **currentSales <= 1000 && newCustomer** |
| --- | --- |
| b. | **currentSales <= 1000 || newCustomer** |
| c. | **currentSales >= 1000 && newCustomer** |
| d. | **currentSales >= 1000 || newCustomer** |

9. Which of the following Boolean expressions tests if a Boolean variable named inactive is not equal to true *and* a decimal variable named netSales is greater than or equal to 0.

| a. | **inactive && netSales >= 0** |
| --- | --- |
| b. | **inactive || netSales >= 0** |
| c. | **!inactive && netSales >= 0** |
| d. | **!inactive || netSales >= 0** |

10. If orderTotal has a value of 50 and quantity has a value of 10, what is the value of discount after these statements are executed?

**if (quantity == 1 || quantity == 2)**

**discount = 0;**

**else if (quantity >= 3 && quantity < 10)**

**discount = orderTotal \* .1;**

**else if (quantity >= 10 && quantity < 25)**

**discount = orderTotal \* .2;**

**else**

**discount = orderTotal \* .3;**

| a. | 15 | c. | 5 |
| --- | --- | --- | --- |
| b. | 10 | d. | 0 |

11. Which of the following statements is *not* true about if statements?

| a. | The statements within the braces of a clause have block scope. |
| --- | --- |
| b. | An if statement can have an unlimited number of else if clauses. |
| c. | Every if statement must include an else clause. |
| d. | You can nest if statements within the if, else if, or else clauses of other if statements. |

12. If quantity has a value of 2, what is the value of discount after these statements are executed?

**switch (quantity)**

**{**

**case 1:**

**discount = 0;**

**break;**

**case 2:**

**case 3:**

**discount = .1;**

**break;**

**default:**

**discount = .2;**

**break;**

**}**

| a. | 0 | c. | .2 |
| --- | --- | --- | --- |
| b. | .1 | d. | 2 |

13. If employeeType has a value of 1, what is the value of employeeDescr after this statement is executed?

**string employeeDescr = employeeType switch**

**{**

**1 => "Management",**

**2 => "Administration",**

**3 => "Marketing",**

**\_ => "Development"**

**};**

| a. | Management | c. | Marketing |
| --- | --- | --- | --- |
| b. | Administration | d. | Development |

14. If employeeType has a value of 5, what is the value of employeeDescr after this statement is executed?

**string employeeDescr = employeeType switch**

**{**

**1 => "Management",**

**2 => "Administration",**

**3 => "Marketing",**

**\_ => "Development"**

**};**

| a. | Management | c. | Marketing |
| --- | --- | --- | --- |
| b. | Administration | d. | Development |

15. If orderTotal has a value of 100 and quantity has a value of 10, what is the value of discount after this statement is executed?

**discount = quantity switch**

**{**

**1 or 2 => 0,**

**>= 3 and < 10 => orderTotal \* .1,**

**>= 10 and < 25 => orderTotal \* .2,**

**\_ => orderTotal \* .3**

**};**

| a. | 15 | c. | 20 |
| --- | --- | --- | --- |
| b. | 10 | d. | 0 |

16. If the GetMessage() method returns “Greetings from Earth!”, what message is displayed after these statements are executed?

**var message = data.GetMessage();**

**if (message is not null && message is string s)**

**{**

**MessageBox.Show(s);**

**}**

**else**

**{**

**MessageBox.Show("No message received.");**

**};**

| a. | “s” |
| --- | --- |
| b. | “Greetings from Earth!” |
| c. | “No message received.” |
| d. | The code throws a null reference exception |

17. Which of the following statements is a more concise way to rewrite the following code?

**decimal salePrice;**

**if (price > 100.0m)**

**salePrice = price \* .7m;**

**else**

**salePrice = price \* .8m;**

| a. | **decimal salePrice = (price > 100.0m) ? price \* .8m : price \* .7m;** |
| --- | --- |
| b. | **decimal salePrice = (price > 100.0m) ? price \* .7m : price \* .8m;** |
| c. | **decimal salePrice = (price > 100.0m) : price \* .8m ? price \* .7m;** |
| d. | **decimal salePrice = (price > 100.0m) : price \* .7m ? price \* .8m;** |

18. What is the value of the variable named counter after the following statements are executed?

**double percent = 0.54;**

**bool valid = true;**

**int counter = 1;**

**if (percent > 0.50 && valid)**

**{**

**counter += 2;**

**if (valid)**

**counter++;**

**else if (percent >= 0.50)**

**counter += 3;**

**}**

**else**

**{**

**counter++;**

**}**

| a. | 2 | c. | 4 |
| --- | --- | --- | --- |
| b. | 3 | d. | 7 |

19. How many times will the statements in the following while loop be executed?

**decimal monthlyInterestRate = .01m;**

**int months = 10;**

**decimal futureValue = 1000m;**

**int i = 1;**

**while (i < months)**

**{**

**futureValue = futureValue \* (1 + monthlyInterestRate);**

**i++;**

**}**

| a. | 10 times | c. | 9 times |
| --- | --- | --- | --- |
| b. | 11 times | d. | None of the above |

20. What is the value of i the first time the while condition is executed at the end of the following do-while loop?

**decimal monthlyInterestRate = .01m;**

**int months = 10;**

**decimal futureValue = 1000m;**

**int i = 1;**

**do**

**{**

**futureValue = futureValue \* (1 + monthlyInterestRate);**

**i++;**

**}**

**while (i < months);**

| a. | 0 | c. | 2 |
| --- | --- | --- | --- |
| b. | 1 | d. | None of the above |

21. What is the value of the futureValue variable after the statements in the following do-while loop have been executed one time?

**decimal monthlyInterestRate = .01m;**

**int months = 10;**

**decimal futureValue = 1000m;**

**int i = 1;**

**do**

**{**

**futureValue = futureValue \* (1 + monthlyInterestRate);**

**i++;**

**}**

**while (i < months);**

| a. | 2010 | c. | 1.01 |
| --- | --- | --- | --- |
| b. | 1010 | d. | None of the above |

22. What is the value of sum after the following code is executed?

**int sum = 0;**

**for (int i = 2; i < 10; i += 2)**

**{**

**sum += i;**

**}**

| a. | 20 | c. | 44 |
| --- | --- | --- | --- |
| b. | 30 | d. | 54 |

23. When you need to increment or decrement a counter variable that determines how many times a loop is executed, you typically want to use

| a. | a while loop |
| --- | --- |
| b. | a do-while loop |
| c. | a for loop |
| d. | an infinite loop |

24. What must you do if you code an infinite loop in an application?

| a. | Use a counter variable to determine when the loop ends |
| --- | --- |
| b. | Use a Boolean expression to determine when the loop ends |
| c. | Code just a single statement within the loop |
| d. | Cancel the application to end the loop |

25. If you code a break statement in a loop, it will cause the application

| a. | to jump to the beginning of the loop | c. | to enter break mode |
| --- | --- | --- | --- |
| b. | to jump to the end of the loop | d. | to restart the loop |

26. If you code a continue statement in a loop, it will cause the application

| a. | to jump to the beginning of the loop | c. | to enter break mode |
| --- | --- | --- | --- |
| b. | to jump to the end of the loop | d. | to jump out of the loop |

27. When you set a breakpoint at a statement and run the application, the application enters break mode

| a. | whenever a runtime error occurs at that statement |
| --- | --- |
| b. | just before the start of the method that contains the statement |
| c. | right after that statement is executed |
| d. | just before that statement is executed |

28. Which of the following can you do in break mode?

| a. | Press F5 to step through the program one statement at a time |
| --- | --- |
| b. | Click the Continue button to step through the program one statement at a time |
| c. | Use the Locals window to review the values of the variables within the scope of the current method |
| d. | Use the Locals window to review the values of all of the variables that the program uses |

29. The first expression that’s coded within the parentheses in a for loop

| a. | increments a counter variable |
| --- | --- |
| b. | decrements a counter variable |
| c. | assigns a starting value to a counter variable |
| d. | tests a Boolean expression |

30. In a do-while loop, the Boolean expression is tested

| a. | before the loop is executed |
| --- | --- |
| b. | after the loop is executed |
| c. | both before and after the loop is executed |
| d. | after the first statement in the loop |

31. In a while loop, the Boolean expression is tested

| a. | before the loop is executed |
| --- | --- |
| b. | after the loop is executed |
| c. | both before and after the loop is executed |
| d. | after the first statement in the loop |

32. What statement is used to implement the selection structure?

| a. | if-else | c. | switch |
| --- | --- | --- | --- |
| b. | while | d. | for |

33. What statement is used to implement the case structure?

| a. | if-else | c. | switch |
| --- | --- | --- | --- |
| b. | while | d. | for |

34. The for and while loops are examples of which structure?

| a. | selection | c. | iteration |
| --- | --- | --- | --- |
| b. | case | d. | match |