|  |  |  |
| --- | --- | --- |
| Name |  |  |
| Score | / 15 |  |  |
| Update Value |  |  |
| Make all corrections and resubmit to earn update points |

 Programming Logic - Beginning

 152-101

 Unit 6

 Repetition Processing

- ¼ point for each incorrect answer, unless specified.

Most answers can be found in Chapter 5 of the book and/or my Unit 6 Instructor’s Notes

**Fill-in-the-Blank**

1. A(n) Click here to enter answer displays a list of items and allows the user to select an item from the list.
2. A(n) Click here to enter answer causes one or more statements to repeat.
3. If a loop does not have a way of stopping, it is called an Click here to enter answer.
4. A(n) Click here to enter answer is a variable that is regularly incremented or decremented each time a loop iterates.
5. Each repetition of the loop is called a(n) Click here to enter answer.
6. A loop that is inside another loop is called a(n) Click here to enter answer loop.
7. Click here to enter answer is the process of inspecting information given to an application by the user and determining whether it is valid.

**Multiple Choice** (highlight the BEST answer)

1. C# automatically adds a \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ to a list box when it contains more items than can be displayed
2. larger list box
3. scroll bar
4. second form
5. message box
6. A ListBox or ComboBox Items numbering starts at
7. 0
8. 1
9. -1
10. any value you specify
11. This property holds the index of the selected item in a list box.
12. Index
13. SelectedItem
14. SelectedIndex
15. Items.SelectedIndex
16. The \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ method erases one item from a list box.
17. Erase
18. Items.Remove
19. Items.RemoveItem
20. Clear
21. The statements between loop curly brackets are known as the \_\_\_\_\_\_\_\_\_\_\_\_\_\_ of the loop.
22. processes
23. functions
24. substance
25. body
26. A(n) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ loop evaluates its test expression before each iteration.
27. out-test
28. pretest
29. posttest
30. in-test
31. The \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ loop is ideal for situations that require a counter because it is specifically designed to initialize, test, and increment a counter variable.
32. Do…While
33. While
34. For
35. Posttest Do Until
36. A \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ has a rectangular area that functions like a Text Box.
37. ListBox
38. DropDown ListBox
39. ComboBox
40. Input Label
41. With this style of ComboBox, the list of items does not drop down, but is always displayed.
42. Drop-down ComboBox
43. Simple ComboBox
44. Drop-down list ComboBox
45. Simple drop-down list ComboBox
46. The ComboBox's \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ property will contain the user's text input or the item selected from the list.
47. Input
48. Caption
49. List
50. Text
51. At design time, the \_\_\_\_\_\_\_\_\_\_\_\_ holds controls that are invisible at runtime, such as the ToolTip control.
52. Component Tray
53. Control container
54. Invisible control box
55. Invisible property

**True or False.** Highlight T or F.

 If the answer is false, write why the answer is false.

|  |  |
| --- | --- |
| T F | 1. The Items.RemoveAt method always removes the last item in a ListBox (the item with the highest index).

Complete if false |
| T F | 1. Infinite loops keep repeating until the program is interrupted.

Complete if false |
| T F | 1. A loop's conditionally executed statements should be indented.

Complete if false |
| T F | 1. A pretest loop always performs at least one iteration, even if the test expression is false from the start.

Complete if false |
| T F | 1. In a For loop, the *counter variable* must be numeric.

Complete if false |
| T F | 1. The *increment* part of the For statement is optional.

Complete if false |
| T F | 1. The For loop is a posttest loop.

Complete if false |
| T F | 1. A drop-down list ComboBox allows the user to either select an item from a list or type text into a text input area.

Complete if false |

**Short Answer** (do only the numbers listed)

1. Write a statement that adds *Spinach* to the list box lstVeggies at index 2.

Click here to enter answer

1. Write a statement that removes the item at index 12 of the ComboBox cmbCourses.

Click here to enter answer

1. In general terms, describe how a While loop works.

Click here to enter answer

1. Why should you indent the statements in the body of a loop?

Click here to enter answer

1. Describe the difference between pretest and posttest loops.

Click here to enter answer

1. Which loop should you use when you know the number of required iterations?

Click here to enter answer

1. Which style of ComboBox forces the user to select an item from the list?

Click here to enter answer

**What do you think?**

1. You use the statement lstNames.Items.RemoveAt(6) to remove an item from a ListBox. Does the statement remove the fifth, sixth or seventh item from the list? Why?

Click here to enter answer

**Find the Errors**

Correct the syntax errors in the following statements.

1. do {

x = x + 1;

}while x < 100

Click here to enter answer

2. do

lstOutput.Items.Add("Hello");

x = x + 1;

while (count < 10);

Click here to enter answer

3. while( x = 99 ) {

x = x + 1;

}//end while

Click here to enter answer

4. for(int x=1, x<=10, x++) {

lstOutput.Items.Add(x);

}//end for

Click here to enter answer

**Algorithm Workbench**

1. An event procedure named btnShow\_Click must add the numbers 1 through 20 to a list box named lstNumbers. Design a flowchart for this event procedure. Use Visio to create the flowchart. When complete, select all the objects (in Visio), copy and paste where designated below

Paste Flowchart Here

1. Write the code for the event procedure described in question 1.

Click here to enter answer

1. Write a For loop that adds the following set of numbers to the list box named lstNumbers.

0, 10, 20, 30, 40, 50 … 1000

Click here to enter answer

1. Convert the following While loop to a For loop.

count = 0

while (count < 50) {

 lstOutput.Items.Add(count);

 count += 1;

}//end while

Click here to enter answer

1. Convert the following For loop to a Do…While loop.

for(int x = 50; x>=0; x--){

lstOutput.Items.Add(x);

}//end for

Click here to enter answer