1. In the program ‘Number Game 2’ which new object was used?
   A ‘ListBox’

2. List some of the different properties of this object?
   Columns, List, Sorted

3. How should this object be named?
   With the prefix lst For example: lstNames

4. Write code to add the contents of a textbox named ‘txtNewname’ to a listbox named ‘lstNames’.
   lstNames.AddItem txtNewname.Text

5. Write code to hide or display a label named ‘lblMessage’ depending on whether it is visible or not.
   If lblMessage.Visible = False Then
     lblMessage.Visible = True
   Else
     lblMessage.Visible = False
   End If

6. Write code to place a random number between 1 and 10000 in a variable named ‘RandomNum1’.
   Randomize
   RandomNum1 = Int(Rnd * 10000 + 1)

7. Why couldn’t the program be changed to accommodate random numbers up to 100 000 in its present form?
   Because the variable SecretNum is an integer variable; it can only hold values up to 32767.

8. What change would have to be made?
   Declare the variable SecretNum as a Long variable:
   Dim SecretNum as Long

9. Write a variable declaration to store integer values of a person’s age.
   Dim Age as Integer

10. Describe two differences between properties and methods.
    Methods can only be used in code in Run Mode; Properties can be set in Design Mode and
        changed in Run Mode.
        Methods don’t use an equals sign to be used in code whereas Properties do.

11. Write code to clear a listbox named ‘lstNames’.
    lstNames.Clear

12. How is the focus set on an object? (Hint: check your code from this project)
    With the SetFocus method, eg, lstNames.SetFocus

13. What is the difference between the ‘Click’ and ‘DoubleClick’ event?
    The Click event is triggered after 1 click on an object. The DoubleClick event is triggered
        after a DoubleClick on an object.

14. Why do you think a ‘DoubleClick’ event is used?
    Because the designer doesn’t want the user to accidentally click and trigger the event.

Extension Activity Notes

The difficulty you would have using ‘RemoveItem’ is to do with whether an item in the list is selected or not. If no
item is selected or no items in the list exist, then the program gives an error on trying to execute the method
‘RemoveItem’. This can be prevented by testing for entries or ensuring an entry is always selected by using a
combination of the ‘ListIndex’ property and the ‘ListCount’ property. Code similar to the following should be
used:

If lstNames.ListCount > 0 And lstNames.ListIndex <> -1 Then  ‘ListIndex = -1 if no item is selected
  LstNames.RemoveItem lstNames.ListIndex
End If