C# Programming: From Problem Analysis to Program Design, 5th edition

Chapter 3

1. b. void

2. d methods

3. c. public

4. c. InputValue

5. b. answer = Math.Floor(87.2);

6. c. int DetermineAnswer(double v1, int v2)

7. a. back to the location in the calling method that made the call

8. c. DetermineHighestScore(val1, val2)

9. e. SetNoOfSquareYards(double)

10. b. local variables

11. d. static int ComputeCost(double aValue)

12. b. call to a void method

13. a. follow the Camel case convention

14. d. InitializeValues( );

15. b. someIntValue = GetData(out aValue, ref bValue);

16. a. heading: void InputValues(out int val1, out int val2)

call: InputValues(out val1, out val2);

17. a. int

18. e. static void DisplayValues(int v1, int v2, int v3)

19. a. static double DetermineGrade(int grade1, int grade2,

int grade3)

20. d. static void DisplayResults(double taxAmount,

double totalSales )

21. a. 2, 2, 0

b. int, void, int

c. All three can have return statements. Only the first and last must have return values.

22. a.

static void DisplayAsterisks( )

{

. WriteLine(“\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*”);

. WriteLine(“\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*”);

. WriteLine(“\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*”);

}

b.

static void DisplayName(string name)

{

WriteLine(“Name: “ + name);

}

c.

static void DisplayNumbers(double v1, double v2)

{

WriteLine(“Value 1 : {0:N3}”, v1);

WriteLine(“Value 2 : {0:N3}”, v2);

}

d..

static int ComputeSum(int v1, int v2, int v3)

{

return (v1 + v2 + v3);

}

23. a. DisplayAsterisks( );

b. DisplayName(“Jason Cheng”);

c. double v1 = 72.87267,

v2 = 98.23;

DisplayNumbers(v1, v2);

d. int ans,

v1 = 782,

v2 = 986,

v3 = 2134;

ans = ComputeSum( v1, v2, v3);

24. a. 64

b. 9

c. −56

25.

Syntax:

1. End the comment with a \*/
2. Syntax error generated in the call to GETVal1( ) method because assignment statement return (val1) does not match variable (VAL1).
3. Syntax error generated in the first call to the WriteLine( ) method because argument (val1) does not match variable (VAL1).
4. Readkey( ) is spelled incorrectly. Should be ReadKey( )
5. Readline( ) is spelled incorrectly. Should be ReadLine( )
6. In order to use WriteLine( ), ReadKey ( ), and ReadLine( ) without qualifying them with the class name, using System.Console; needs to be added
7. aValue could be declared as a string argument in GetAge( ) method. No need to send it as an argument. Could send age as an argument. If you did, it would need to have the out parameter type added to both the method heading and the call.
8. Need a return type for the GETVal1 method heading -static int GETVal1 ( )
9. In the GETVal1( ) method, need to declare a local variable for age
10. Return statement for the GETVal1( ) method should not have int

Style Issues

-Variable VAL1 identifier shown in upper case should be camel case.

-GETVal1( ) method name also does not follow standard naming convention. Name of the method should be GetVal1( ). Should change the heading and the call to the method.

- For readabillty, indent the statements in the Main( ) method and match the curly braces.

Below is the working solution with the above modifications.

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

using System;

using System.Console;

namespace ErrorExample

{

class ErrorExample

{

static void Main( )

{

int val1;

string aValue;

val1 = GetVal1( );

WriteLine("Value entered, plus one is {0}",

++val1);

ReadKey( );

}

static int GetVal1( )

{

int val1;

string aValue;

Write("Enter your age: ");

aValue = ReadLine( );

val1 = int.Parse(aValue);

return val1;

}

}

}