C# Programming: From Problem Analysis to Program Design, 4th edition

ISBN 978-1-285-09626-1

Chapter 3

1. d. methods

2. d void

3. a. private

4. d. PrintReport

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

6. c. public 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. public 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: public void InputValues(out int val1, out int val2)

call: InputValues(out val1, out val2);

17. a. int

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

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

int grade3)

20. d. public 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.

public static void DisplayAsterisks( )

{

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

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

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

}

b.

public static void DisplayAge(int age)

{

Console.WriteLine(“Age: “ + age);

}

c.

public static void DisplayNumbers(double v1, double v2)

{

WriteLine(“Value 1 : {0:N3}”, v1);

WriteLine(“Value 2 : {0:N3}”, v2);

}

d.

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

{

return (v1 + v2 + v3);

}

e.

public static void PrintBoolean( )

{

bool aValue = true;

Console.WriteLine(“Boolean value: “ + aValue);

}

23. a. 98887.234

b. 64

c. -1

d. 9

e. -56

24.

- End the comment with a \*/ on line 6

- Variable age identifier shown in upper case (style issue).

-Syntax error generated in the first call to the WriteLine( ) method because argument (age) does not match variable (AGE).

-GETAge( ) method name also does not follow standard naming style convention.

- 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.

- Need a return type for the GetAge method heading -- public static int GetAge( )

- Name of the method should be GetAge( ) [style issue]. Should change the heading and the call to the method.

- In the GetAge( ) method, need to declare a local variable for age

- return for the GetAge( ) should not have int

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

Below is the working solution with the above modifications.

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

\* AgeIncrementer.cs Author: Doyle

\* Prompts the user for their age in

\* a class method. Prints the value

entered back in the Main( ) method.

\*/

using System;

namespace AgeExample

{

public class AgeIncrementer

{

public static void Main( )

{

int age;

string aValue;

age = GetAge( );

Console.WriteLine("Your age next year will be {0}",

++age);

Console.Read( );

}

public static int GetAge( )

{

int age;

string aValue;

Console.Write("Enter your age: ");

aValue = Console.ReadLine( );

age = int.Parse(aValue);

return age;

}

}

}