

In the space provided, answer the following 4 questions. The test is marked out of 25.

1) [6 marks] Multiple Choice: for each question, circle the best answer.

i) A parameter within a method header is called a(n) ____ parameter.

- ☒ a. formal
- ☐ b. local
- ☐ c. global
- ☐ d. actual

ii) Methods are ____ when there is the potential for a situation in which the compiler cannot determine which method to use.

- ☒ a. overloaded
- ☐ b. ambiguous
- ☐ c. polymorphic
- ☐ d. hidden

iii) In order to use a Binary Search, the array must ____.

- ☐ a. contain integers
- ☐ b. not be negative
- ☒ c. be sorted
- ☐ d. be passed by value

iv) When one user-defined method calls another user-defined method, the calls are ____.

- ☒ a. nested
- ☐ b. interleaved
- ☐ c. cascaded
- ☐ d. complex

v) ____ describe(s) a situation in which the compiler cannot determine which method to use.

- ☒ a. Overloading
- ☐ b. Ambiguous methods
- ☐ c. Polymorphism
- ☐ d. Confusion

vi) Any optional formal parameters in a parameter list must appear to the right of all ____ formal parameters in the list.

- ☐ a. local
- ☒ b. mandatory
- ☐ c. default
- ☐ d. global

vii) C# supports a ____ statement that you can use to cycle through every array element without using a subscript.

- ☐ a. for
- ☐ b. while
- ☐ c. do-while
- ☒ d. foreach

viii) A ____ statement causes a value to be sent back to the calling method.

- ☐ a. break
- ☐ b. goto
- ☒ c. return
- ☐ d. continue

ix) Only ____ parameters can be given default values.

- ☒ a. reference
- ☐ b. output
- ☐ c. value
- ☐ d. named

x) Using a(n) ____ formal parameter is convenient when the variable to be passed does not have an assigned value at the time the method is called.

- ☒ a. reference
- ☐ b. value
- ☐ c. mandatory
- ☐ d. output

xi) You can distinguish each element from the others in an array with a(n) ____.

- ☐ a. element operator
- ☐ b. object location
- ☐ c. compass
- ☒ d. subscript

xii) A ____ array is a one-dimensional array in which each element is another one-dimensional array with a possibly different length.

- ☐ a. rectangular
- ☐ b. matrix
- ☒ c. jagged
- ☐ d. table

2) [4 marks]

(a) Convert the following segment of code so that it uses a foreach loop.

```
double [] averageMark = new double[6] { 76.4, 81.2, 65.9, 92.6, 88.8, 70.0 };
for (int i = 0; i < averageMark.Length; ++i)
    if(averageMark[i] > 80)
        Console.WriteLine("On the Dean's List");
    else
        Console.WriteLine("Try harder ... you can do it!");
```

(b) Convert the following segment of code so that it uses a *for* loop.

```
int[] studentNumber = new int[6] { 1234, 2345, 6463, 8263, 2814, 5431 };
foreach (int stID in studentNumber)
    if(stID < 5000)
        Console.WriteLine("Wow, you have been here a long time");
    else
        Console.WriteLine("You must have just started");
```

3) [6 marks] Consider the following C# program with **two** user-defined methods?

```
using System;
public static class Quest3
{
    public static void Main()
    {
        int a = 9;
        Console.WriteLine("Hello from Main");
        Method2(ref a, -1);
        Console.WriteLine("Welcome back: the value is {0}", a);
        Console.ReadLine();
    }
    public static int Method1(int b, int c = 4)
    {
        Console.WriteLine("Hello from Method1");
        return c+1;
    }
    public static void Method2(ref int x, double y, int z = 7)
    {
        Console.WriteLine("Hello from Method 2");
        x = Method1(c: 3, b: 0);
        return;
    }
}
```

54

a) What are the formal parameters? <i>int b, int c = 4</i> <i>ref int x, double y, int z = 7</i>	d) What are the call by value formal parameters? <i>int c = 4, int z = 7</i> <i>b, y</i>
b) What are the actual parameters (arguments)? <i>ref a, -1</i> <i>c: 3, b: 0</i>	e) What are the optional parameters? <i>int c = 4, int z = 7</i>
c) What are the named arguments? <i>c: 3, b: 0</i>	f) What are the call by reference formal parameters? <i>ref int x</i>

4) [4 marks] What is the output of the following program?
using System;

```
public class Quest4
{
    public static void Main()
    {
        Console.Write("W"); /
        F1();
        Console.WriteLine("I");
    }

    public static void F1()
    {
        Console.Write("N"); 2
        F3();
    }

    public static void F2(int x)
    {
        if (x < 0)
            Console.Write("T");
        else
        {
            F2(-1);
            Console.Write("E");
        }
    }

    public static void F3()
    {
        F2(0);
        Console.Write("R");
    }
}
```

5) [5 marks] Write a C# method called **Smallest** which takes three **int** formal parameters (**x1**, **x2**, and **x3**) and returns the an integer indicating which one is the smallest. That is, **Smallest** is to return a -1 if **x1** is the smallest, 0 if **x2** is the smallest, and 1 if **x3** is the smallest. You are ONLY to write the method, that is, DO NOT WRITE Main(). The method is not to do any input or output: you can assume that those functions are performed in Main(). The method heading should look like:

```
public static int Smallest(int x1, int x2, int x3)
```