Student Name: Todd Tracy

|  |  |  |
| --- | --- | --- |
| IT FDN using C# | EXAM 2 | Instructor:  Vallejo |

Note:  The test is worth **100 points**.  **Show all your work** for each problem.  No partial credit will be given if no work is shown for each answer.  Read the entire description to each question before answering the question.  **Good Luck!**

**True / False (2 points)**

***Circle One***

1.  Array indices start at one. TRUE / FALSE

2.  Array.Rank is the total number of elements. TRUE / FALSE

3.  Array.Length is the number of array dimensions. TRUE / FALSE

4.  **foreach** is used to iterate through an array. TRUE / FALSE

5.  **protected** class data is used in inheritance. TRUE / FALSE

6.  Static variables retain their values for the life of the program. (if you mean they retain the same value and the value can not be changed then FALSE, if you mean the value can change but can be used through many different classes of the program then TRUE) TRUE / FALSE

7.  Constructors are used to initialize class data. TRUE / FALSE

8.  Accessor functions are used to hide data. TRUE / FALSE

9. The elements of an array can be different types. TRUE / FALSE

10. The elements of a structure can be different types. TRUE / FALSE

**Multiple Choice (3 points)**

11.  Which is the correct operator to access a member of a structure?

      A.   **.**

      B.   **[ ]**

      C.   **( )**

      D.   **!**

12. Several functions with the same name are called:

      A.   overall

      B.   oversize

      C.   overloading

      D.   overdone

13. If aiArray has 10 elements, which is the last logically valid accessible element:

      A.   aiArray[8]

      B.   aiArray[9]

      C.   aiArray[10]

      D.   aiArray[11]

14. Which is the correct way to declare a two-dimensional array:

      A.   int[ , ] aiArray;

      B.   int[ ] aiArray;

      C.   int[ ][ ] aiArray;

      D.   int[ ].[ ] aiArray;

15. The parameter to a function **int AddSum(int iVal)** is:

      A.   passed by ref

      B.   passed by value

      C.   passed by pointer

      D.   None of the above

16. A local variable's scope is:

      A.   within a module

      B.   within a function

      C.   within a statement

      D.   None of the above

17. Declare an integer array of size 100:

      A.  int[] numbers = new int[100];

      B.  int numbers = new[] int[100];

      C.  int[100] numbers = new int[];

      D.  int new numbers = int[100];

18. Properties should have the following:

      A.   let / set

      B.   get / set

      C.   get only

      D.   set only

19. Which is the correct way to test two strings for equality?

      A.   Str1 == Str2

      B.   \*Str1 == \*Str2

      C.   &Str1 == &Str2

      D.   None of the above

20. Which one is a correct way to access a method from class Point:

      A.   point.MyMethod();

      B.   Point.MyMethod();

      C.   class point.MyMethod()

      D.   class MyMethod();

**Short Answer**

21. What happens when you create an object of a class? Briefly describe the steps that happen behind the scene to the class that we instantiate from. (5 points)

Well a class is basic blueprint for classifying objects. So the object is the instantiate of a class. Like a class is a recipe for different type of cookies and the objects are the actual cookie.

22.  What will the following display? (5 points)

using System;

class Test

{

static void Main( )

{

int[] X = new int[10] {0,1,4,9,16,0,0,0,0,0};

int   k;

for (k = 5; k < 10; ++k) // k = 5 as it 5 and increments up by 1 after, then repeats in loop until k is 10.

   {

      X[k] = k \* k;

      }

for (k = 0; k < X.Length; k++) //Then moves to this and resets k = 0 so:

   {

      Console.Write("{0}    ", X[k]);

      }

}

}

**Display = 0    1    4    9    16    0    36    49    64    81    100**

23. What will the following do? (5 points)

using System;

class Factorial

{

    public static void Main()

    {

        long nFactorial = 1;

        long nComputeTo = 5;

        long nCurDigit = 1;

        try

        {

            long x = 1 / (1 - nFactorial); //Since 1-1 = 0 the code errors out right here as you can’t divide 1 by 0.

            checked //lets say it makes it past that it errors out.

            {

                for (; nCurDigit <= nComputeTo; nCurDigit++)//nCurDigit will increase to 6 as ComputeTo equals 5.

                {

                    nFactorial **= nCurDigit;** // Every time nCurDigit increases by 1 it will multiple my the itn of nFactorial and replace nFactorial (1 \* 1 = 1, 1\*2=2, 2\*3 =6, 6\*4=24, 24\*5=120. At nCurDigit = 6 it breaks out of the loop and goes below.

                }

            }

        }

        catch (OverflowException e)

        {

            Console.WriteLine("Computing {0}! caused an overflow {1}",

                nComputeTo, e.StackTrace);

            return;

        }

        Console.WriteLine("{0}! is {1}", nComputeTo, nFactorial); //Should right something like “5! is 120”

    }

}

**Behavior =** see above

24.  What will the following display? (5 points)

using System;

class Shape

{

}

class Test

{

   static void Main()

   {

      Shape s = new Shape();

      Console.WriteLine(s);

   }

}

**Display = Shape**

25.  Define a structure that contains a student name, social security number, number of classes taken, and a letter grade. (5 points)

public struct student

{public string name;

public int SSN;

public int number\_of\_classes;

public string letter\_grade;

}

26.  Define an **enum** for the seasons (Summer, Spring, Winter, and Fall). (5 points)

 public enum seasons

{

Summer,

Spring,

Winter,

Fall,

}

**Problem Solving**

27.  Given the following program what will be displayed. (see below) Is there anything unusual about this program? That is subjective. I have added notes on what this program does. (10 points)

using System;

class Test

{

    static void Main()

    {

        int[] xlist = new int[] {9,5,3,-2,4,5}; // xlist is an array int[6]

        for (int x = 0; x < xlist.Length; x++) // Does through each number in the xlist until it hits int [2] which is 3

        {

            if (xlist[x] == 3)

            {

                for (int y = x; y < xlist.Length - 1; y++)

                {

                    xlist[y] = xlist[y+1]; //This copies the information from the next element in the array to the current element so 3 becomes -2, -2 becomes 4, and 4 becomes 5.

                }

            }

        }

        foreach (int v in xlist)

        {

            Console.Write("{0} ", v); //Since the for statement didn’t reduce the total amount of elements in the array there is still 6 elements so the int 5 repeats in element 5 and 6.

        }

    }

}

Display = 9, 5, -2, 4, 5, 5

28.  Given the following program what will be displayed. (see below)  Is there anything unusual about this program? If flipping the elements in the array is unusual then yes, if not then no. (10 points)

using System;

class Test

{

    static void Main()

    {

        int[] xlist = new int[] {7,-2};

        for (int x = 0; x < xlist.Length - 1; x++)

        {

            if (xlist[x] > xlist[x+1])

            {

                int t = xlist[x]; //saving element number 7.

                xlist[x] = xlist[x+1]; //changing 7 to -2.

                xlist[x+1] = t; //replacing -2 with 7.

            }

        }

        foreach (int v in xlist)

        {

            Console.Write("{0} ", v);

        }

    }

}

**Display = -2, 7**