Student Name:

|  |  |  |
| --- | --- | --- |
| 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

C# arrays are zero indexed i.e. the array indexes start at zero.

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

Gets the number of dimensions.

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

Gets the total number of elements.

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

foreach statement repeats the given statements for each element in the array.

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

6.  Static variables retain their values for the life of the program. TRUE / FALSE

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

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

Accessor functions are used to provide access to the user but not directly the variables declared.

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

Elements of array must be of single type. Array of objects can be different types.

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.   **!**

Example:

struct student

{

public string name;

public int age;

}

class Test

{

static void Main()

{

student[] students = new student[2];

students[0].name = "Steve";

students[0].age = 1234567;

……

}}

12. Several functions with the same name are called:

      A.   overall

      B.   oversize

      C.   overloading

      D.   overdone

Example

static void number(int x, int y)

{

…

}

static void number(int x, double y)

{

…

}

static void number(double x, int y)

{

…

}

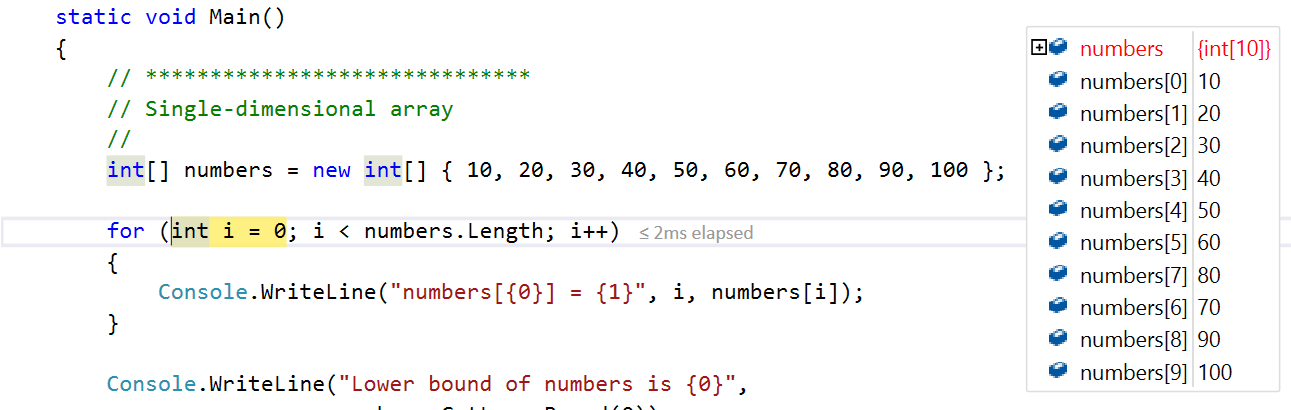
13. If aiArray has 10 elements, which is the last logically valid accessible element:

      A.   aiArray[8]

      B.   aiArray[9]

      C.   aiArray[10]

      D.   aiArray[11]

 Example: 

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

      A.   int[ , ] aiArray;

      B.   int[ ] aiArray;

      C.   int[ ][ ] aiArray;

      D.   int[ ].[ ] aiArray;

// Two-dimensional array.

int[,] array2D = new int[,] { { 1, 2 }, { 3, 4 }, { 5, 6 }, { 7, 8 } }

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

The scope of a local variable declared in a switch-block of a switch statement is the switch-block.

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

 Property is a block of code that allows us to access a class data fields in a safe and easy way. A property is accessed via accessor functions get and set.

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();

 Precise example in the Answer to question 21 (below). Where GetArea method is called.

**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)

 Answer: Using an example to explain.

using System;

namespace GetArea

{

class Rectangle

{

private double length;

private double width;

//constructor for a rectangle class that accepts two parameters and which are initialized.

public Rectangle(double l, double w)

{

length = l;

width = w;

}

// Method to calculate the area of a rectangle and the GetArea will be used when called from else where in the code.

public double GetArea()

{

return length \* width;

}

}

class Program

{

static void Main()

{

Rectangle rect = new Rectangle(10.0, 20.0); // Object is created using the 'new' keyword followed by the call to the class constructor.

double area = rect.GetArea();

Console.WriteLine("Arear of an Rectangle: {0}", area);

Console.ReadLine();

/\* When the code executes and object of Rectangle type is created in the memory.

A reference to this memory is stored inside the rect. Using the object reference the class members are accessed.

Code here calls on GetArea method on the object and the values returned by the method is stored in the area.

Lenght and Width of the object rect cannot be accessed directly as they are marked private.

\*/

}

}

}

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)

   {

      X[k] = k \* k;

      }

for (k = 0; k < X.Length; k++)

   {

      Console.Write("{0}    ", X[k]);

      }

}

}

**Display = \_\_\_\_**

**0**

**1**

**4**

**9**

**16**

**25**

**36**

**49**

**64**

**81\_\_\_\_\_**

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)

            checked

{

for (; nCurDigit <= nComputeTo; nCurDigit++)

{

nFactorial \*= nCurDigit;

                }

            }

        }

catch (OverflowException e)

        {

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

                nComputeTo, e.StackTrace);

            return;

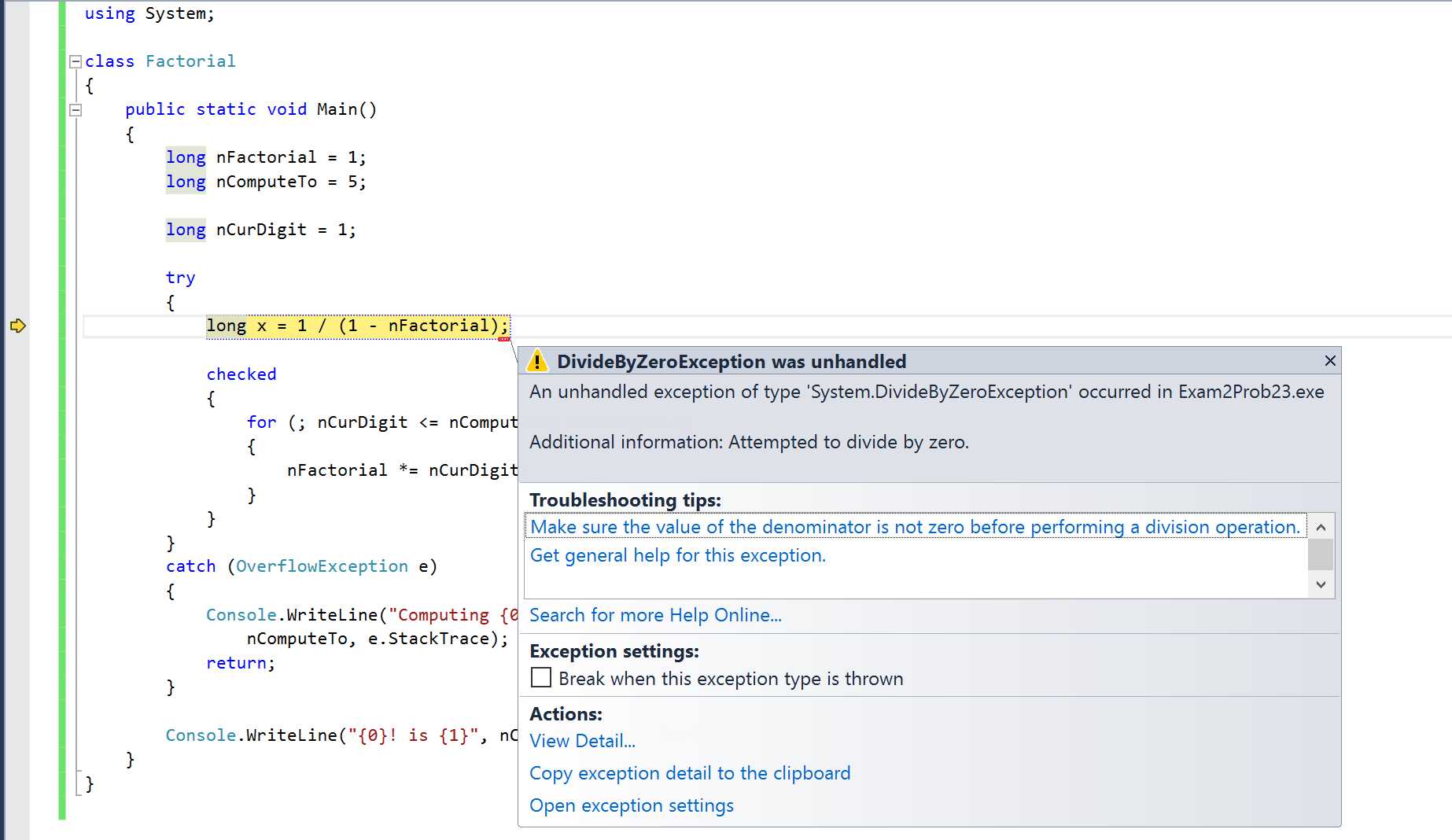
        }

        Console.WriteLine("{0}! is {1}", nComputeTo, nFactorial);

    }

}

**Behavior = \_**When executed it threw an unhandled exception (screenshot below) as the denominator of the of the division operation will result in zero **\_**



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)

Answer

struct student

{

public string studentname;

public int ssn;

public int classestaken;

public string grade;

}

class Test

{

static void Main()

{

student[] students = new student[2];

students[0].studentname = "Steve";

students[0].ssn = 1234567;

students[0].classestaken = 2;

students[0].grade = "B";

students[1].studentname = "Dan";

students[1].ssn = 098765;

students[1].classestaken = 6;

students[1].grade = "A";

foreach (student p in students)

{

Console.WriteLine("{0} has taken {1} number of classes.", p.studentname, p.classestaken);

Console.ReadLine();

}

}

}

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

 Answer:

using System;

enum SeasonNames

{

Summer = 1,

Spring,

Winter,

Fall

}

class Test

{

static void Main()

{

SeasonNames season;

season = SeasonNames.Winter;

if (season == SeasonNames.Summer)

{

Console.WriteLine("Yeah, its a holiday time!");

}

else if (season == SeasonNames.Spring)

{

Console.WriteLine("Yeah, Its colorful time!");

}

else if (season == SeasonNames.Winter)

{

Console.WriteLine("Its cold!");

}

Console.WriteLine("Season is {0} and its integer value is {1}",

season, (int)season);

Console.Write("Press enter to close the window........");

Console.ReadLine();

}

}

P**roblem Solving**

27.  Given the following program what will be displayed.  Is there anything unusual about this program? (10 points)

using System;

class Test

{

    static void Main()

    {

        int[] xlist = new int[] {9,5,3,-2,4,5};

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

        {

            if (xlist[x] == 3)

            {

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

                {

                    xlist[y] = xlist[y+1];

                }

            }

        }

        foreach (int v in xlist)

        {

            Console.Write("{0} ", v);

        }

    }

}

Display = \_\_\_

9

5

-2

4

5

5\_\_\_

28.  Given the following program what will be displayed.  Is there anything unusual about this program? (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];

                xlist[x] = xlist[x+1];

                xlist[x+1] = t;

            }

        }

        foreach (int v in xlist)

        {

            Console.Write("{0} ", v);

        }

    }

}

**Display = \_\_\_\_\_\_**

**-2**

**7\_\_\_\_\_\_\_**

Execution of code works perfectly fine. No errors or exceptions, not able to find anything unsual when executed line by line. If there is anything that a beginner brain is not able to pick. Please help understand if there is anything unusual 😊.