Trent University

Computing and Information Systems 1020H

Sample Final Examination

Examiner: Dr. R.T. Hurley   Duration: 3 hours

NAME:___________________________   **STUDENT#** _______________________

OPEN BOOK – Course Notes/Textbook

In the space provided, answer the following 10 questions. The exam is graded out of 60.

1) [8 marks] Answer the following questions either True (T) or False (F):

a) An object should be encapsulated, guarding its data from inappropriate access. ____T___

b) Identifier in C# may contain letters, numbers, and any punctations marks. ____F___

c) The left hand side of an assignment statement must be a variable or constant. ____T___

d) Properties do not allow validations and computations before accepting a value or returning a value. ____F___

e) A method that receives an array as a parameter cannot permanently change elements of the array. ____F___

f) If possible, parameters are converted to conform with their arguments. ____F___

g) The impact of changes made to a parameter inside the method depends on the type of the parameter. ____T___

h) The versions of an overloaded method are distinguished by the return type, number, type, and order of their parameters. ____F___

i) To pass an array by address, the argument and parameter must use *ref*. ____F___

j) Because the || operator performs short-circuit evaluation, your Boolean expression will be evaluated faster if the sub-expression that is most likely to be true is on the left. ____T___

k) A *for* loop can always be replaced with a *while* loop. ____T___

l) C# does not limit the number of dimensions that an array may have. ____T___

m) A sorting algorithm is used to locate a specific item in a larger collection of data. ____F___

n) In order to operate on a 2D array, two *for* loop are required. ____F___

o) A static method can access an instance field (variable). ____F___

p) The public access specifier for an attribute indicates that the attribute may not be accessed by statements outside the class. ____F___

2) [4 marks] What is the output of the following program?

```
using System;
public class Quest2
{
     public static void Main()
     {
          int x, y;
          x = Convert.ToInt32(Console.ReadLine());
          y = Convert.ToInt32(Console.ReadLine());
          if((2 * x) > y)
             if(y > x)
                  Console.WriteLine("Loc 1");
             else if(x < 0)
                  Console.WriteLine("Loc 2");
             else
                  Console.WriteLine("Loc 3");
          else
             Console.WriteLine("Loc 4");
     }
 }
```

Assume the following input is used (program is re-run for each set of input)

a) 3 4

Loc 1

b) 6 3

Loc 3

c) -2 5

Loc 4

d) -1 -3

Loc 2

3) [5 marks] Write a C# method to determine the registration fee for automobiles. The method *Registration* should take two parameters, *year* (int) and *weight* (double), and returns a double representing the registration fee based on the following table:

Model Year	**Weight**	**Registration Fee**
1980 or earlier	less than 2000 kg	$34.50
	greater than or equal to 2000 kg	58.75
Greater than 1980	less than or equal to1500 kg	45.25
	greater than 1500 kg	75.00

You are ONLY to write the method (DO NOT WRITE the main program). The method heading should look like:

```
public static double Registration(int year, double weight)
{
if (year <= 1980)
     if (weight < 2000)
        return 34.50;
     else
        return 58.75;
else
     if (weight <= 1500)
        return 45.25;
     else
        return 75.00;
}
```

4) [5 marks] What is the output of the following program?

```
using System;
public class Quest4
{
    public static void Main()
    {
        Console.Write("H");
        F3();
        Console.WriteLine("U");
    }

    public static void F1(int y)
    {
        F2(0);
        Console.Write("R");
    }

    public static void F2(int x)
    {
        if (x > 1)
            Console.Write("L");
        else
            {
            Console.Write("E");
            F2(2);
            }
    }

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

ANSWER

HYELRU

5) [5 marks] What is the output of the following program (i.e. what is outputted by the Console.WriteLine statement as the program executes). As we did in class, use a table to keep track of the variable values through the loop.

```
using System;
public class Quest5
{
    public static void Main()
    {
        int a = 0, x, y, z = 0;
        while (a < 5)
        {
            x = Convert.ToInt32(Console.ReadLine());
            y = Convert.ToInt32(Console.ReadLine());
            if(x > y)
                z += x;
            else
                z−−;
            Console.WriteLine("a = {0},  x = {1}, y = {2}, z = {3}", a, x, y, z);
            a += 2;
        }
    }
}
```

Given the input:

5	2
3	6
4	4
7	5

ANSWER

a = ___0__, x = ___5__, y = ___2__, z = ___5__

a = ___2__, x = ___3__, y = ___6__, z = ___4__

a = ___4__, x = ___4__, y = ___4__, z = ___3__

a = ______, x = ______, y = ______, z = ______

6) [3 marks] Convert the following *for* loop into a *while* loop.

```
for( int x = 0, int y = 5 ; x <= y; x++, y−=2)
{
     Console.WriteLine("x = {0}, y = {1}", x, y);
}

     int x = 0;
     int y = 5;
     while (x <= y)
     {
          Console.WriteLine("x = {0},y = {1}", x, y);
          x++;
          y −= 2;
     }
```

7) [5 marks] What is the output of the following program?

```
using System;
public class Quest6
{
     public static void Main()
     {
          int a = 3, b = 5, c = 7;
          Console.WriteLine("a = {0}, b = {1}, c = {2}", a, b, c);
          c = fun1(ref b, a);
          Console.WriteLine("a = {0}, b = {1}, c = {2}", a, b, c);
     }
     public static int fun1(ref int x, int y)
     {
          Console.WriteLine("x = {0}, y = {1}", x, y);
          x = y / 2;
          fun2(ref  y);
          Console.WriteLine("x = {0}, y = {1}", x, y);
          return 2;
     }
     public static void fun2(ref int w)
     {
          w = -1;
     }
}
```

ANSWER

```
a = ___3__, b = ___5__, c = __ 7___
x = ___5__, y = ___3__
x = ___1__, y = __-1__
a = ___3__, b = ___1__, c = ___2___
```

8) [5 marks] Consider the following function called *PercentFreq* which takes two parameters: an integer array *x* and a integer *target*. The function is to determine the number of times that *target* appears in the array *x,* and then returns the percentage of time it appeared in the array. For example, if *target* was found 3 times and the array *x* had 9 elements, then the method should return 33.3 (i.e. 3/9). Fill in the missing blanks to make the function work.

```
public ______(a)______ double PrecentFreq (int [] x, int _____(b)_____)
    {
    int count = 0;
    for(int i = 0; i < x.Length; ++i)
        {
        if( ________(c)_______ == target)
            ________(d)_______;
        }
    return ________(e)_______;
    }
```

(a)	static
(b)	target
(c)	x[i]
(d)	count++
(e)	count / (double) x.Length

9) [5 marks] What is the output from running the following program that uses arrays (ie., what will appear in the output window after running the program):

```
using System;
public static class Quest9
{
    public static void Main()
    {
        int [] array1 = new int [4] {7, -2, 0, 6};
        int [] array2 = new int [4] {0, 0, 0, 0};

        for(int i = 0; i < array2.Length; i++)
        {
           if(array1[i] >= 0)
                array2[i] = array1[i] + 2;
           else
                array2[i] = array1[i] * (-1);
        }

        Console.Write("Array 1: ");
        for (int i = 0; i < array2.Length; ++i)
           Console.Write("{0} ", array1[i]);
        Console.WriteLine();

        Console.Write("Array 2: ");
        for (int i = array2.Length-1; i >= 0; i--)
           Console.Write("{0} ", array2[i]);
        Console.WriteLine();

        Console.ReadLine();
    }
}
```

ANSWER

```
Array 1:  7  -2  0  6
Array 2:  8  2  2  9
```

10) [15 marks] Consider the following Class Definition for a program that allows the user to work with either Fahrenheit of Celsius temperatures:

```
using System;
public class Temperature
    {
    private double degrees;
    private char scale;         //'F' or 'C'
    public  Temperature();      // no arg constructor
    {
       degrees = 0;
       scale = 'C';
    }
    public Temperature(float deg, char which)   // two arg constructor
    {          // insert code here            }
    public double Degrees                       // Property
    {             // insert code here         }
    public char Scale                           // Property
    {             // insert code here         }
    public override string ToString()
    {             // insert code here         }
    private double ConvertToCel()
    {             // insert code here         }
    public static operator+ (Temperature temp1, Temperature temp2)
    {             // insert code here         }
};
```

a) Give the details for the constructor Temperature(float deg, char which) which should initialize each instance fields to *deg* and *which*. Be sure to ensure scale is either 'F' or 'C'

```
public Temperature(float deg, char which)
{
    degree = deg;
    if (which == 'C' || which == 'F')
       scale = which;
    else
       scale = 'C';
}
```

b) Give the details for the ToString() member function which simply rerurns a string specifying the current temperature and scale (e.g. “32 degrees F”).

```
public override string ToString()
{
    string str;
    str = Degrees + “ degrees “ + Scale;
    return str;
}
```

c) Give the details for gets and sets of the Degrees or Scale property. Be sure to validate any data before changing the instance field. Just complete the code for one of these properties.

```
public double Degrees
{
    get     {
            return degrees;
            }
    set     {
            degrees = value;
            }
}
```

```
// or public char Scale
    {
        get     {
                return degrees;
                }
        set     {
                if(value == ‘F’ || value == ‘C’)
                    scale = value;
                else
                    scale = ‘C’;
                }
    }
```

d) Give the details for the ConvertToCel() method. The method will convert the current temperature stored in the *degrees* field of the object to Celsius and then return the value. The method does not change the value stored in the *degrees* field (it simply determines a Celsius equivalent of it). If the current value is already in Celsius, simply return the *degrees* field. In case you do not remember, the equation for conversion is:

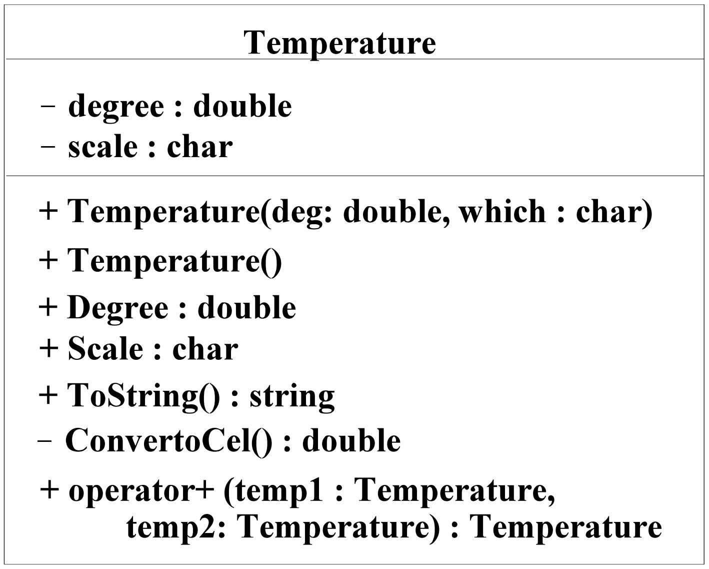
Cel = (Far - 32) * 5/9

```
private double ConvertToCel()
{
    if (scale == 'F')
        {
        val = (Degrees - 32.0) * 9.0 / 5 .0;
        return val;
        }
    else
        return Degrees;
}
```

e) Give the details for the same class method Add(). In order to add the values of the two temperatures, they must be the same scale. Use the ConvertToCel() method to convert any temperature that is in Fahrenheit. Your Add() method is to return a new object which stores the sum of the other two temperatures in Celcius.

```
public static operator+ (Temperature temp1, Temperature temp2)
{
    Temperature temp3 = new Temperature();
    double sum;
    if(temp1.Scale == 'F')
        temp1.Degrees = temp1.ConvertToCelcius();
    If(temp2.Scale == 'F')
        temp2.Degrees = temp2.ConvertToCelcius();
    temp3.Degrees= temp1.Degrees + temp2.Degrees;
    return temp3;
}
```

f) Draw the UML diagram for the Temperature class.



```
                Temperature
--------------------------------------------------
 − degree : double
 − scale : char
--------------------------------------------------
 + Temperature(deg: double, which : char)
 + Temperature()
 + Degree : double
 + Scale : char
 + ToString() : string
 − ConvertoCel() : double
 + operator+ (temp1 : Temperature,
        temp2: Temperature) : Temperature
```