CS127T Object Oriented Programming.

Assignment #3

Name :Syed Obaid Hashmi
Roll NO:2020fCs-030

[Question-1]

a) IndexOutOfRangeException:

CODE:

```
//An IndexOutOfRangeException exception is thrown when an invalid index is used to access a member of an array or a collection, or to read or write from a particular location in a buffer. This exception inherits from the Exception class but adds no unique members

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApplication20

{

    class withexception
    {
        public void insert_data_in_array()
        {
            int[] array = new int[3];

            try
            {
                array[0] = 22;
                array[1] = 23;
                array[2] = 44;
                array[3] = 73;
                array[4] = 87;
                foreach (int i in array)
                {
                    Console.WriteLine(i);
                }
            }
            catch(IndexOutOfRangeException ex)
            {
                Console.WriteLine(" Error has occurred : "+ex.Message);
```

```
            }

        }

    }

    class Program
    {
        static void Main(string[] args)
        {

            withexception we = new withexception();
            we.insert_data_in_array();

            Console.ReadLine();
        }
    }
       }
```

(b) ArgumentOutOfRangeException:

CODE:

```
        // An ArgumentOutOfRangeException exception is thrown when a method is invoked and

        at least one of the arguments passed to the method is not null and contains an

        invalid value that is not a member of the set of values expected for the argument.

using System;


class Pinfo
{
    private string FirstName;
    private string LastName;
    private int Age;

    public Pinfo(string fName, string lName, int age)
    {
        FirstName = fName;
        LastName = lName;
        if (age < 21)
            throw new ArgumentOutOfRangeException("age", "All guests must be 21-years-old
or older.");
        else
            Age = age;
    }

    public string display()
    {
        string display = FirstName + " " + LastName + ", " + Age.ToString();
        return (display);
    }
```

```
}


class Program
{
    static void Main(string[] args)
    {
        try
        {
            Pinfo p1 = new Pinfo("obaid", "hashmi", 17); //Error Occurred here! Age less
than 21 years..
            Console.WriteLine(p1.display());
        }
        catch (ArgumentOutOfRangeException outOfRange)
        {

            Console.WriteLine("Error:" + outOfRange.Message);
        }
        Console.ReadLine();
    }
}
```

(C) ArrayTypeMismatchException:

CODE:

```
//. ArrayTypeMismatchException is thrown when the system cannot convert the element

to the type declared for the array. For example, an element of type String cannot

be stored in an Int32 array because conversion between these types is not supported.

using System;

namespace ConsoleApplication20
{
    class Arraymis
    {
        static void Main(string[] args)
        {
            string[] names = { "mobile", "Car", "cycle" };
            Object[] objs = (Object[])names;

            try
            {
                objs[2] = "Bus";

                foreach (object objName in objs)
                {
                    Console.WriteLine(objName);
                }
            }
            catch (System.ArrayTypeMismatchException)
            {
                // Not reached
```

```
            Console.WriteLine("Exception Thrown.");
        }

        try
        {
            Object obj = (Object)13;
            objs[2] = obj;
        }
        catch (System.ArrayTypeMismatchException)
        {

            Console.WriteLine(
                "New element is not of the correct type.");

        }

        Console.ReadLine();
    }
}
   }
```

(d) OutofMemoryException:

CODE:

```
//An OutOfMemoryException exception has two major causes: You are attempting to
expand a StringBuilder object beyond the length defined by its StringBuilder.
MaxCapacity property. The common language runtime cannot allocate enough contiguous
memory to successfully perform an operation.

using System;
using System.Text;
public class Program
{
    public static void Main()
    {
        StringBuilder stringBuilder = new StringBuilder(17, 17);
        stringBuilder.Append("Welcome to the ");
        try
        {
            stringBuilder.Insert(0, "world of C# programming", 1);
            Console.WriteLine(stringBuilder.ToString());
            Console.ReadLine();
        }
        catch (OutOfMemoryException exception)
        {
            Console.WriteLine(exception.Message);
            Console.ReadLine();
        }
    }
```

```
file:///C:/Users/HASHMI/documents/visual studio 2012/Projects/ConsoleApplication20/ConsoleApplication20/bin/Debug/ConsoleApplication20.EXE
Insufficient memory to continue the execution of the program.
```

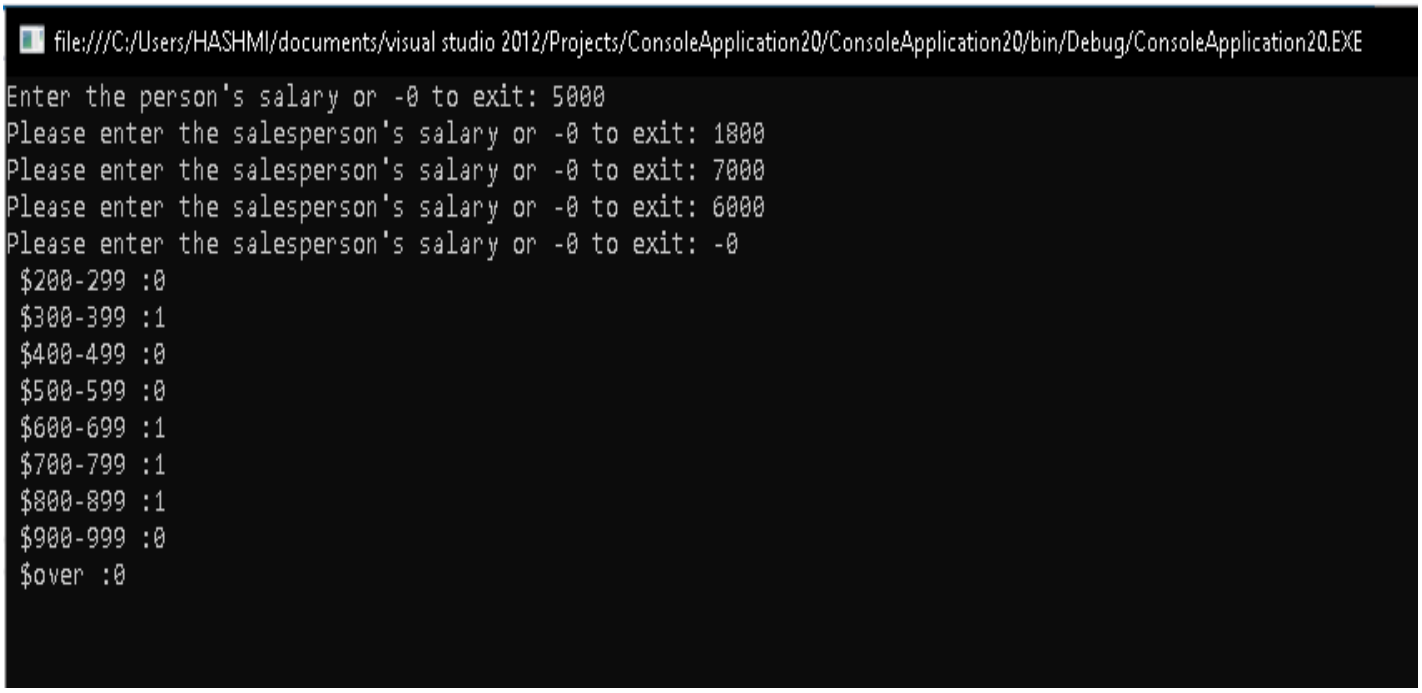
[Question-2]

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApplication20
{
    class Program
    {
        static void Main(string[] args)
        {
            double[] array = new double[9];

            Console.Write("Enter the person's salary or -0 to exit: ");
            double iValue = Convert.ToDouble(Console.ReadLine());

            while (iValue != -0)
            {
                iValue = (iValue * (0.09)) + 200;//For example, a salesperson who
grosses(ivalue)$1800 in sales in a week receives $200 plus 9% of $1800, or a total of
$362.

                if (iValue < 300)
                {
                    array[0] = array[0] + 1;}

                     else if (iValue < 400)
                    {
                        array[1] = array[1] + 1;}

                      else if (iValue < 500)
                        {
                            array[2] = array[2] + 1;}


                          else if (iValue < 600)
                            {
                                array[3] = array[3] + 1;}

                              else if (iValue < 700)
                                {
                                    array[4] = array[4] + 1;}

                                 else  if (iValue < 800)
                                   {
                                       array[5] = array[5] + 1;}

                                    else if (iValue < 900)
                                     {
                                         array[6] = array[6] + 1;}
```

```
                                        else if (iValue < 1000)
                                         {
                                             array[7] = array[7] + 1;
                                         }

                                           else
                                              {
                                                array[8] = array[8] + 1;
                                              }

                                    Console.Write("Please enter the salesperson's salary
or -0 to exit: ");
                                    iValue = Convert.ToInt32(Console.ReadLine());
                                                }

            for (int i = 0; i < array.Length; i++)
            {
                switch (i)
                {
                    case 0:
                        Console.WriteLine(" $200-299 :" + array[i]);
                        break;

                    case 1:
                        Console.WriteLine(" $300-399 :" + array[i]);
                        break;

                    case 2:
                        Console.WriteLine(" $400-499 :" + array[i]);
                        break;

                    case 3:
                        Console.WriteLine(" $500-599 :" + array[i]);
                        break;

                    case 4:
                        Console.WriteLine(" $600-699 :" + array[i]);
                        break;

                    case 5:
                        Console.WriteLine(" $700-799 :" + array[i]);
                        break;

                    case 6:
                        Console.WriteLine(" $800-899 :" + array[i]);
                        break;

                    case 7:
                        Console.WriteLine(" $900-999 :" + array[i]);
                        break;

                    case 8:
                        Console.WriteLine(" $over :" + array[i]);
                        break;
                }
            }
            Console.ReadLine();
        }
```

```
    }
}
```

[OUTPUT]


file:///C:/Users/HASHMI/documents/visual studio 2012/Projects/ConsoleApplication20/ConsoleApplication20/bin/Debug/ConsoleApplication20.EXE
Enter the person's salary or -0 to exit: 5000
Please enter the salesperson's salary or -0 to exit: 1800
Please enter the salesperson's salary or -0 to exit: 7000
Please enter the salesperson's salary or -0 to exit: 6000
Please enter the salesperson's salary or -0 to exit: -0
$200-299 :0
$300-399 :1
$400-499 :0
$500-599 :0
$600-699 :1
$700-799 :1
$800-899 :1
$900-999 :0
$over :0