Module 2 Practicals

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

namespace ConsoleApp1
{
    class Program
    {
        static void Main(string[] args)
        {
            /* ----------------Part-1-------------------*/
            int A = 10;
            int B = 20;

            //arithmetic operators
            int res = 0;
            res = A + B;
            Console.WriteLine("Addition: "+res);
            res = A - B;
            Console.WriteLine("Subtraction: "+res);
            res = A * B;
            Console.WriteLine("Multiplication: "+res);
            res = B / A;
            Console.WriteLine("Division: "+res);
            res = B % A;
            Console.WriteLine("Modulo: "+res);
            res = A++;
            Console.WriteLine("Post Increment A: "+res);
            res = ++A;
            Console.WriteLine("Pre Increment A: "+res);
            res = B--;
            Console.WriteLine("Post Decrement B: "+res);
            res = --B;
            Console.WriteLine("Pre Decrement B: "+res);

            //relational operators
            bool res1;
            res1 = A == B;
            Console.WriteLine("A==B: "+res1);
            res1 = A != B;
            Console.WriteLine("A!=B: "+res1);
            res1 = A > B;
            Console.WriteLine("A>B: "+res1);
            res1 = A < B;
            Console.WriteLine("A<B: "+res1);
```

```
                res1 = A >= B;
                Console.WriteLine("A>=B: "+res1);
                res1 = A <= B;
                Console.WriteLine("A<=B: "+res1);

                //logical operators
                bool a = true;
                bool b = false;
                res1 = a && b;
                Console.WriteLine("And: "+res1);
                res1 = a || b;
                Console.WriteLine("Or: " + res1);
                res1 = ! (a || b) ;
                Console.WriteLine("Not: " + res1);

                //bitwise operators
                A = 60;
                B = 13;
                res = A & B;
                Console.WriteLine("Bitwise And: "+res);
                res = A | B;
                Console.WriteLine("Bitwise Or: "+res);
                res = A ^ B;
                Console.WriteLine("Bitwise Xor: "+res);
                res = ~A;
                Console.WriteLine("Bitwise 1's Complement: "+res);
                res = A << 2;
                Console.WriteLine("Bitwise Shift Left: "+res);
                res = A >> 2;
                Console.WriteLine("Bitwise Shift Right: "+res);

//assignment operator-every arithmetic and bitwise operators can be used as follows:
                //res += A;

                //misc operators
                res = sizeof(int);
                Console.WriteLine("Size of Integer: "+res);
                res = A > B ? 100 : 200;
                Console.WriteLine("Ternary Operator: "+res);

                Console.ReadKey();


                /* ---------------Part-2------------------*/
                int i;

                //for loop
                for(i=0 ; i<5 ; i++)
                    Console.WriteLine(i);
```

```
//while loop
while(i > 0)
{
        Console.WriteLine(i);
        i--;
}

//do-while loop
do
{
        Console.WriteLine(i);
        i++;
} while (i < 5);

//for-each loop
int []x = { 1, 2, 3, 4, 5 };
foreach(int item in x)
        Console.WriteLine(item);

//continue control stmt
for (i = 0; i < 5; i++)
{
        if (i == 3)
                continue;
        Console.WriteLine(i);
}

//break control stmt
for (i = 0; i < 5; i++)
{
        if (i == 3)
                break;
        Console.WriteLine(i);
}

//goto control stmt
Console.WriteLine("Enter your age:");
int age = Convert.ToInt32(Console.ReadLine());
if (age < 18)
        goto ineligible;

else
{
        Console.WriteLine("You are eligible to vote!");
        goto exit;
}


ineligible:
        Console.WriteLine("You are not eligible to vote!");
```

```
exit:

//infinite loop
/*for (i = 0; ; i++)
{
      Console.WriteLine(i);
}

for (; ;)
{
      Console.WriteLine("Trapped");
}
*/
Console.ReadKey();



/* ----------------Part-3-------------------*/
//single dimensional array
int []n = new int[10];
int i;

for (i = 0; i < 10; i++)
{
      n[i] = i + 100;
}

foreach (int m in n)
{
      i = m - 100;
      Console.WriteLine("Element[{0}] = {1}", i, m);
}

//multi dimensional array
int [,]a = new int[5, 2] { { 0, 0 }, { 1, 2 }, { 2, 4 }, { 3, 6 }, { 4, 8 } };
int j;

for (i = 0; i < 5; i++)
{
      for (j = 0; j < 2; j++)
      {
            Console.Write("a[{0},{1}] = {2} ", i, j, a[i, j]);
      }
      Console.WriteLine();
}

//jagged array
int [][]b = new int[][]{new int[]{0,0},new int[]{1,2},new
int[]{2,4},new int[]{ 3, 6 }, new int[]{ 4, 8 } };
```

```
            for (i = 0; i < 5; i++)
            {
                for (j = 0; j < 2; j++)
                {
                    Console.Write("a[{0}][{1}] = {2} ", i, j, b[i][j]);
                }
                Console.WriteLine();
            }

            //passing array as parameter
            double avg;

            avg = getAverage(n,10);
            Console.WriteLine("Average value is: {0} ", avg);

            //param array
            int sum = AddElements(512, 720, 250, 567, 889);
            Console.WriteLine("The sum is: {0}", sum);

            //array class
            Object o = 100;

            Console.WriteLine("Array Length Property: "+n.Length);
//ans in 32 bit integer
            Console.WriteLine("Array LongLength Property: " + n.LongLength);
//ans in 64 bit integer
            Console.WriteLine("Array Rank Property: " + a.Rank);
//dimensions of array

            Console.WriteLine("Array GetLength Method: " + n.GetLength(0));
//ans in 32 bit integer
            Console.WriteLine("Array GetLongLength Method: " +
n.GetLongLength(0));    //ans in 64 bit integer
            Console.WriteLine("Array GetValue Method: " + n.GetValue(1));
//value at given index
            Console.WriteLine("Array IndexOf Method: " + Array.IndexOf(n,o));
//first occurance of obj in array

            Console.WriteLine("Array Reverse Method: ");
            Array.Reverse(n);
//reverse array
            foreach (int m in n)
            {
                Console.Write(m + " ");
            }
            Console.WriteLine();

            Console.WriteLine("Array SetValue Method: ");
            n.SetValue(o,1);
//set obj value at given index
```

```
            foreach (int m in n)
            {
                Console.Write(m + " ");
            }
            Console.WriteLine();

            Console.WriteLine("Array Sort Method: ");
            Array.Sort(n);
//sort array
            foreach (int m in n)
            {
                Console.Write(m + " ");
            }
            Console.WriteLine();

            Console.WriteLine("Array ToString Method: ");
            n.ToString();
//convert array to string
            foreach (int m in n)
            {
                Console.Write(m + " ");
            }
            Console.WriteLine();

            Console.WriteLine("Array Copy Method: ");
            int[] n1 = new int[20];
            Array.Copy(n,n1,10);
//copy one array to another
            foreach (int m in n1)
            {
                Console.Write(m + " ");
            }
            Console.WriteLine();

            Console.WriteLine("Array CopyTo method: ");
            n.CopyTo(n1,                                   10);
//copy one array to another from given index
            foreach (int m in n1)
            {
                Console.Write(m + " ");
            }
            Console.WriteLine();

            Console.WriteLine("Array Clear method: ");
            Array.Clear(n,0,10);
//clear array
            foreach (int m in n)
            {
                Console.Write(m + " ");
            }
```

```
 Console.WriteLine();
 Console.ReadKey();

/* ----------------Part-4------------------*/
 //all methods are declared in "methods" class

 //method declaration and usage
 int a = 100;
 int b = 200;
 methods o = new methods();

 int res = o.add(a , b);
 Console.WriteLine("Result: "+res);

 //passing parameters by value
 Console.WriteLine("Before swapping,a : {0}", a);
 Console.WriteLine("Before swapping,b : {0}", b);

 o.swap(a, b);

 Console.WriteLine("After swapping,a : {0}", a);
 Console.WriteLine("After swapping,b : {0}", b);

 //passing parameters by reference
 Console.WriteLine("Before swapping,a : {0}", a);
 Console.WriteLine("Before swapping,b : {0}", b);

 o.swap(ref a, ref b);

 Console.WriteLine("After swapping,a : {0}", a);
 Console.WriteLine("After swapping,b : {0}", b);

 //passing parameters by output
 o.add(a, b, out res);
 Console.WriteLine("Result: "+res);

 //passing parameters by optional parameters
 o.scholar("Heli","Parekh");
 o.scholar("Deepika","Padukone",39);
 o.scholar("Rohan","Parekh",30,"Information Technology");

 Console.ReadKey();

 /* ----------------Part-5------------------*/
            //calling different constructors
            char[] a = { 'H', 'e', 'l', 'i', ' ', 'P', 'a', 'r', 'e', 'k', 'h' };
            string org = "Hello";
            string s1 = org;//copy constructor
            Console.WriteLine(s1);
            string s2 = new string(a);//joining character array
            Console.WriteLine(s2);
            string s3 = new string(a, 2, 5);//creating a substring
            Console.WriteLine(s3);
            string s4 = new string('c', 3);//joins 3 times the character
specified
            Console.WriteLine(s4);

            //creating string using different ways
            string fname, lname;
            fname = "Heli";
            lname = "Parekh";

            string[] sarray = { "Hello", "Everyone" };

            string fullname = fname + " " + lname;//concatenation
            Console.WriteLine("Full Name: {0}", fullname);

            string message = String.Join(" ", sarray);//join method
            Console.WriteLine("Message: {0}", message);

            DateTime time = new DateTime(2012, 10, 10, 17, 58,
1);//formatting
            string chat = String.Format("Message sent at {0:t} on
{0:D}", time);
            Console.WriteLine("Time: {0}", chat);

            //Properties
            Console.WriteLine("Char Property: " + s4[2]);//chars
property
            Console.WriteLine("Length Property: " +
s4.Length);//gives length of string

            //methods
            char[] b = { 'a', 'e', 'r', 'p' };
            char[] c = new char[5];

            Console.WriteLine("Compare Method: " +
String.Compare("abc", "Abc"));
            Console.WriteLine("Compare Method: " +
String.Compare("abc", "ABC", true));//ignores case-true
            Console.WriteLine("CompareTo Method: " +
s1.CompareTo("hello"));//compare s1 and hello
            Console.WriteLine("Concat Method: " +
String.Concat(fname, lname));
            Console.WriteLine("Concat Method: " +
String.Concat(fname, " ", lname));
            Console.WriteLine("Contains Method: " +
s1.Contains("H"));//s1 contains "H" or not
            Console.WriteLine("Copy Method: " +
String.Copy(s1));//copies s1
```

```
            s2.CopyTo(0, c, 0, 5);
            Console.WriteLine("CopyTo Method: "); //copies 'heli '
from s2 to char array c
            for (int i = 0; i < c.Length; i++)
                Console.Write(c[i]);
            Console.WriteLine();

            Console.WriteLine("Equals Method: " +
s2.Equals(fullname));//compares s2 and fullname
            Console.WriteLine("EndsWith Method: " +
s2.EndsWith("h"));//s2 endswith "h" or not
            Console.WriteLine("StartsWith Method: " +
s2.StartsWith("H"));
            Console.WriteLine("IndexOf Method: " +
s2.IndexOf('h'));
            Console.WriteLine("IndexOfAny Method: " +
s2.IndexOfAny(b));
            Console.WriteLine("LastIndexOfAny Method: " +
s2.LastIndexOfAny(b));
            Console.WriteLine("Substring Method: " +
s2.Substring(2));
            Console.WriteLine("Insert Method: " + s1.Insert(5, "
Everyone"));
            Console.WriteLine("Replace Method: " + s1.Replace('e',
'E'));
            Console.WriteLine("Remove Method: " + s1.Remove(3));
            Console.WriteLine("ToLower Method: " +
s1.ToLower());
            Console.WriteLine("ToUpper Method: " +
s1.ToUpper());
            Console.WriteLine("Trim Method: " + s2.Trim());

            //string builder
            StringBuilder sb = new StringBuilder("Heli", 10);
            //StringBuilder s = new StringBuilder("Heli");
            //StringBuilder s = new StringBuilder(10);
            //StringBuilder s = new StringBuilder();
            sb.Append(" Parekh");
            sb.AppendLine(" 14");//new line added after 14
            sb.Append("Hello Everyone");
            Console.WriteLine(sb);

            StringBuilder sb1 = new StringBuilder("Amount: ");
            sb1.AppendFormat("{0:C} ", 50);
            Console.WriteLine(sb1);

            sb.Insert(21, ",Hi!!");
            Console.WriteLine(sb);
```

```
                        sb.Remove(21, 5);
                        Console.WriteLine(sb);

                        sb.Replace("Hello", "Hi");
                        Console.WriteLine(sb);

                        Console.ReadKey();

      /* ----------------Part-6-------------------*/
                        //various constructors
                        DateTime DOB = new DateTime(1956, 12, 8, 6, 24, 12,
23);// From DateTime create the Date and Time
                        Console.WriteLine(DOB);

                        string DateString = "8/12/1956 7:10:24 AM";// From
String creation of DateTime
                        DateTime dateFromString =
DateTime.Parse(DateString,
System.Globalization.CultureInfo.InvariantCulture);
                        Console.WriteLine(dateFromString);

                        DateTime EmpDateTime = new DateTime();// Empty
DateTime
                        Console.WriteLine(EmpDateTime);

                        DateTime OnlyDate = new DateTime(2020, 10, 19);//
Just date
                        Console.WriteLine(OnlyDate);

                        DateTime OnlyTime = new DateTime(10000000);//
DateTime from Ticks
                        Console.WriteLine(OnlyTime);

                        DateTime DateTimewithKind = new DateTime(1976, 7,
10, 7, 10, 24, DateTimeKind.Local);// Localization with DateTime
                        Console.WriteLine(DateTimewithKind);

                        //properties
                        Console.WriteLine("Day:{0}", DOB.Day);
                        Console.WriteLine("Month:{0}", DOB.Month);
                        Console.WriteLine("Year:{0}", DOB.Year);
                        Console.WriteLine("Hour:{0}", DOB.Hour);
                        Console.WriteLine("Minute:{0}", DOB.Minute);
                        Console.WriteLine("Second:{0}", DOB.Second);
                        Console.WriteLine("Millisecond:{0}",
DOB.Millisecond);
                        Console.WriteLine("Day of Week:{0}",
DOB.DayOfWeek);
                        Console.WriteLine("Day of Year: {0}",
DOB.DayOfYear);
```

```
                Console.WriteLine("Time of Day:{0}",
DOB.TimeOfDay);
                Console.WriteLine("Ticks:{0}", DOB.Ticks);
                Console.WriteLine("Today:{0}", DateTime.Today);
                Console.WriteLine("Now:{0}", DateTime.Now);
                Console.WriteLine("UTC:{0}", DateTime.UtcNow);
                Console.WriteLine("Kind:{0}",
DateTimewithKind.Kind);

                //methods
                TimeSpan Month = new System.TimeSpan(30, 0, 0, 0);
                DateTime Day = DateTime.Now;
                DateTime SubtractDate = new DateTime(2000, 2, 6, 13, 5,
15);

                DateTime aDayAfterAMonth = Day.Add(Month);//add
30 days to current day
                Console.WriteLine(aDayAfterAMonth);

                DateTime aDayBeforeAMonth =
Day.Subtract(Month);//subtract 30 days(timespan) from current day
                Console.WriteLine("{0:dddd}", aDayBeforeAMonth);
                TimeSpan Difference =
DOB.Subtract(SubtractDate);//subtract another date
                Console.WriteLine(Difference);

                Console.WriteLine(Day.AddYears(2));//add various
components to datetime
                Console.WriteLine(Day.AddDays(12));
                Console.WriteLine(Day.AddHours(4.25));
                Console.WriteLine(Day.AddMinutes(15));
                Console.WriteLine(Day.AddSeconds(45));
                Console.WriteLine(Day.AddMilliseconds(200));
                Console.WriteLine(Day.AddTicks(5000));

                int NumberOfDays = DateTime.DaysInMonth(2020, 2);
                Console.WriteLine(NumberOfDays);//number of days in
a year can be found using same technique in for loop for 12 months of a year

                DateTime DateOfFirst = new DateTime(2002, 10, 22);
                DateTime DateOfSecond = new DateTime(2009, 8, 11);
                int result1 = DateTime.Compare(DateOfFirst,
DateOfSecond);//compare two dates
                if (result1 < 0)//result1=-1
                    Console.WriteLine("Date of First is earlier");
                else if (result1 == 0)
                    Console.WriteLine("Both dates are same");
                else//result1=1
                    Console.WriteLine("Date of First is later");
Console.WriteLine(DateOfFirst.CompareTo(DateOfSecond));//same work as
above

                    /*
                    string[]                    FormatsOfDate                    =
OnlyDate.GetDateTimeFormats();//to get formats available
                    foreach (string format in FormatsOfDate)
                         Console.WriteLine(format);
                    FormatsOfDate = OnlyDate.GetDateTimeFormats('d');//
DateTime Formats: d, D, f, F, g, G, m, o, r, s, t, T, u, U
                    foreach (string format in FormatsOfDate)
                         Console.WriteLine(format);
                    Console.WriteLine(OnlyDate.ToString("r"));//formates
can be specified as such
                    */



Console.WriteLine(OnlyDate.IsDaylightSavingTime());//to      check      for
daylight saving time

Console.WriteLine(DateTime.IsLeapYear(OnlyDate.Year));//to   check   leap
year

                    Console.WriteLine("ToString:              "             +
DOB.ToString());//conversions of datetime
                    Console.WriteLine("ToBinary: " + DOB.ToBinary());
                    Console.WriteLine("ToFileTime:             "            +
DOB.ToFileTime());
                    Console.WriteLine("ToLocalTime:            "            +
DOB.ToLocalTime());
                    Console.WriteLine("ToLongDateString:         "          +
DOB.ToLongDateString());
                    Console.WriteLine("ToLongTimeString:         "          +
DOB.ToLongTimeString());
                    Console.WriteLine("ToOADate: " + DOB.ToOADate());
                    Console.WriteLine("ToShortDateString:         "         +
DOB.ToShortDateString());
                    Console.WriteLine("ToShortTimeString:         "         +
DOB.ToShortTimeString());
                    Console.WriteLine("ToUniversalTime:          "          +
DOB.ToUniversalTime());

                    Console.ReadKey();

  }

     //methods for part-3
 public static double getAverage(int[] arr, int size)
```

```
        {
            int i;
            double avg;
            int sum = 0;

            for (i = 0; i < size; ++i)
            {
                sum += arr[i];
            }
            avg = (double)sum / size;
            return avg;
        }

        public static int AddElements(params int[] arr)
        {
            int sum = 0;

            foreach (int i in arr)
            {
                sum += i;
            }
            return sum;
        }
    }

    //methods for part 4
    class methods
        {
            public int add(int a,int b)
            {
                return a + b;
            }

            public void swap(int x, int y)
            {
                int temp;

                temp = x;
                x = y;
                y = temp;
            }

            public void swap(ref int x, ref int y)
            {
                int temp;

                temp = x;
                x = y;
                y = temp;
            }
```

```
                public void add(int a, int b , out int res)
                {
                     res = a + b;
                }

                public void scholar(string fname,string lname,int age = 21,string
branch = "Computer Engineering")
                {
                     Console.WriteLine("First name: {0}", fname);
                     Console.WriteLine("Last name: {0}", lname);
                     Console.WriteLine("Age: {0}", age);
                     Console.WriteLine("Branch: {0}", branch);
                }

           }

}
```