Thursday, February 3, 2022.

C# OOP's Concept
(constructors ,class ,object).

By Nalli_Prudhvi.
Nations_Benifits.

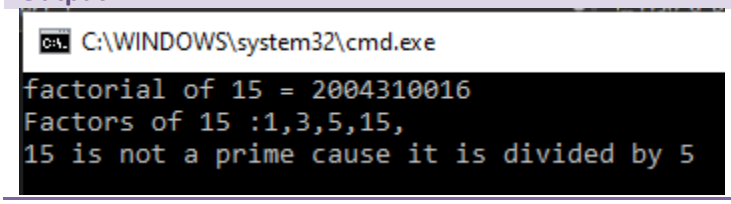
Q. Create mathematical operation class write methods to find factors, factorial, and prime check.

Code

```
class Quick_math
    {
        public int z;

        public Quick_math(int i)
        {
            z = i;
        }
        /// <summary>
        /// prints the factorial of input
        /// </summary>
        public void Factorial()
        {
            int fact = z;
            for(int i = 1; i < z; i++)
            {
                fact *= i;
            }
            Console.WriteLine($"factorial of {z} = {fact}");

        }
        /// <summary>
        /// prints factors of input
        /// </summary>
        public void Factors()
        {
            Console.Write($"Factors of {z} :");
            for(int f = 1;f<= z; f++)
            {
                if(z%f == 0)
                    Console.Write(f+",");
            }
            Console.WriteLine();
        }
        /// <summary>
        /// print the number Is prime
        /// </summary>
        public void Isprime()
        {
            bool cn=true;
            int check = z-1;
            while(check>=2)
            {
                if (z % check == 0)
                {
                    cn = false;
                    break;
                }
                else
                    check--;
            }
            if (cn)
                Console.WriteLine($"{z} is a Prime number");
            else
```

```
                Console.WriteLine($"{z} is not a prime cause it is divided by
{check}");
            }
        }
        internal class Program
        {
            static void Main(string[] args)
            {
/*****************************************************************************
 *Author : Nalli_Prudhvi                                                     *
 *Purpose: Create mathematical operation class write methods to find factors, *
 *       factorial, and prime check                                          *
 *****************************************************************************/

                var obj1 = new Quick_math(15);
                obj1.Factorial();
                obj1.Factors();
                obj1.Isprime();
                Console.ReadLine();
            }
        }
```

Output

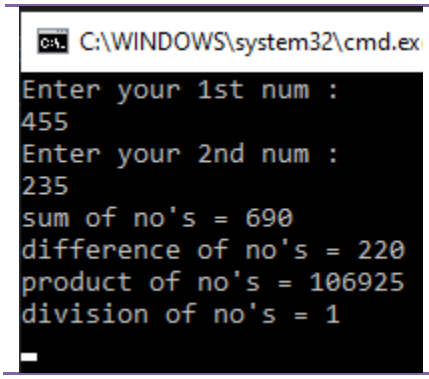


Q. Create math operator class and write print values of sum, Difference, product, division

Code

```
class Quick_math2
    {
        public int z;
        public int x;
        public Quick_math2(int z, int x)
        {
            this.z = z;
            this.x = x;
        }
        /// <summary>
        /// Prints the values mathematical op. b/w two no's.
        /// </summary>
        public void Math_op()
        {
            int sum = z + x;
            int sub = z - x;
            int mul = z * x;
            int div = z / x;
            Console.WriteLine($"sum of no's = {sum}");
            Console.WriteLine($"difference of no's = {sub}");
            Console.WriteLine($"product of no's = {mul}");
            Console.WriteLine($"division of no's = {div}");
        }

    }
    internal class Program
    {
        static void Main(string[] args)
        {
/*************************************************************************
 *Author : Nalli_Prudhvi                                                 *
 *Purpose: Create math operator class and write print values of sum,     *
 *      Difference, product, division.                                   *
 *************************************************************************/

            Console.WriteLine("Enter your 1st num :");
            int a = Convert.ToInt32(Console.ReadLine());
            Console.WriteLine("Enter your 2nd num :");
            int b = Convert.ToInt32(Console.ReadLine());
            var ob1 = new Quick_math2(a, b);
            ob1.Math_op();
            Console.ReadLine();
        }
    }
```

Output

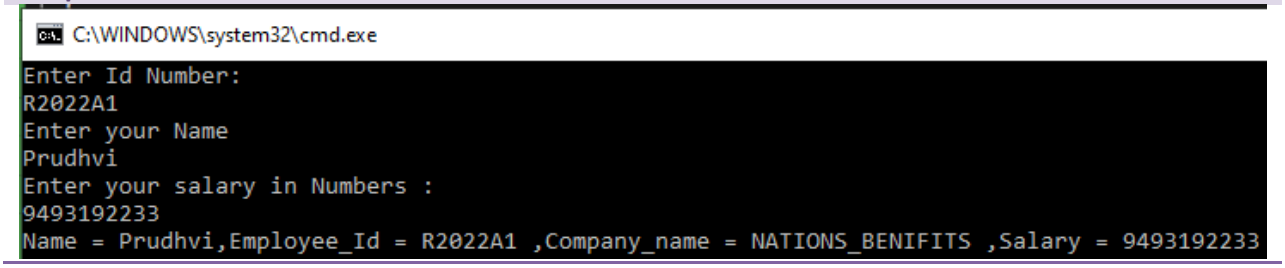

C:\WINDOWS\system32\cmd.ex
Enter your 1st num :
455
Enter your 2nd num :
235
sum of no's = 690
difference of no's = 220
product of no's = 106925
division of no's = 1

Q. Create Employee class written methods to read and write.

Code

```
class Employee
    {
        public string Id;
        public string Name;
        public static string Company_name = "NATIONS_BENIFITS";
        public long Salary;

        public void Employee_Data()
        {
            Console.WriteLine("Enter Id Number:");
            Id = Console.ReadLine();
            Console.WriteLine("Enter your Name");
            Name = Console.ReadLine();
            Console.WriteLine("Enter your salary in Numbers :");
            Salary = Convert.ToInt64(Console.ReadLine());
        }
        public void Print_employee_data()
        {
            Console.WriteLine($"Name = {Name},Employee_Id = {Id} ,Company_name =
{Company_name} ,Salary = {Salary}  ");
        }
    }

    internal class Program
    {
        static void Main(string[] args)
        {
/*****************************************************************************
 *Author : Nalli_Prudhvi                                                     *
 *Purpose: Create Employee class written methods to read and write data     *
 *****************************************************************************/
            var emp_1 = new Employee();
            emp_1.Employee_Data();
            emp_1.Print_employee_data();
        }
    }
```

Output



Q. Difference between Normal and Static Variable.	
Normal variable	Static Variable
Normal variables can be accessed using instance of a class	Static variables can be accessed using class name.
Normal variables cannot be accessed inside a static method	Static variables can be accessed by static and normal methods.
Normal method do not reduce the memory used.	Static method reduce the memory used.
Normal variables are used in same instance of a class	Static variables are shared among all instance.

Q. 5 Points about constructor.

A. **A Constructor is used to initialize class variables while creating an object.**
B. **By default, we will have default constructor which will initialize to default values.**
C. **The moment the programmer create user define constructor the default constructor will be gone.**
D. **We can also construct a default constructor with the user define constructor, for default value so, it will not show a error , if we leave empty.**
E. **If your using same variables in constructor variables as that of the class variables, use this. to differentiate with the class variables.**
EX: this.id = id

Q. create a class read var with constructor and print.

Code

```
class Employee
    {
        public string Id;
        public string Name;
        public static string Company = "Nations_Benifits";
        public long Salary;

        public Employee(string Id, string Name, long Salary)
        {
            this.Id = Id;
            this.Name = Name;
            this.Salary = Salary;
        }
        public Employee()
        {
            Id = null;
            Name = null;
            Salary = 0;
        }
        public void Print_data()
        {
            Console.WriteLine($"Employee_name = {Name}, Employee_Id = { Id},
Employee_company = {Company}, Employee_salary = {Salary}");
        }

    }
    internal class Program
    {
        static void Main(string[] args)
        {/*************************************************************************
          * Author  :Nalli_Prudhvi                                                *
          * Purpose :create a class read var with constructor and print data   *
          * ***********************************************************************/
            Employee emp = new Employee("Az0022ON","Jeff",9441319223);
            emp.Print_data();
            Console.ReadLine();
        }
    }
```

Output