JUNAID MALIK

C# NOTES

0304-1659294

MUHAMMAD JUNAID

AL-JUNAID TECH INSTITUTE | junaidfazal08@gmail.com

History of C#

- It is an Object Oriented Programming Language
- It is Fastest Language
- It is Apply on .NET framework(.NET is SEGMENT of MICROSOFT. It is a Product of MICROSOFT)
- You can C# Use to create windows application, web Servers, Mobile Applications, Clients server Application, Database Applications, and Many More
- C# developed in 2000
- It is Brain of Gaming Industry

History of .NET Framework

- The .NET Framework Consist of the Common Language Runtime (CLR: it execute the CODE. It is also Called IDE.) and the .NET Frame work class Library
- F#, C# and Visual Basic are the Base of .NET
- The CLR is the Foundation of the .NET Framework. It Manages the code at execution time, providing core services such as memory management, code accuracy and many other aspects of your code.
- Code manage, Memory allocation and check every aspect CLR manage all of these.

USES of C#:

- C# is used by Programming to develop computer, software, games, website and web app
- It is used to create general System Software
- It is Used to build drivers for Various Computer Devices

- It is also Used in 3D Modeling (auto cate or auto desk or Maya) and Sculptures like UNITY (like as Gamming engine) and Visual Studio.

Advantages of C#:

- C# is relatively-high level and is a System Programming Language Based on .NET Framework
- It has a Large Community
- It has a relatively clear and Mature Standard
- Modularity
- Reusability and readability

OUR FIRST PROGRAM:

```
//USING is a preprocessing commond all Statmenr declaration
Line are save in preprocessing Commond
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

// namespace it is ELEMNTS Like as Class objects or Statment
are seperate
namespace JUNAID_MALIK
{
    // The Main Function of namespace is ignore the SPACE
    class Program  //Class its Stor the Multiple objects
    {
        //Main function (declair or Predefault start from Main
function
        static void Main(string[] args)
        {
            //Console.WriteLine means PRINT
            Console.WriteLine("Hello : JUNAID MALIK");
            Console.ReadLine(); //Out put Hold krne ke lie
        }
    }
}
```

AL-JUNAID INSTITUTE OF GROUP

DATA TYPE:

- INT
- CHAR
- DOUBLE
- BOLEAN
- STRING

```
//USING is a preprocessing commond all Statmenr declaration
Line are save in preprocessing Commond
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

// namespace it is ELEMNTS Like as Class objects or
Statment are seperate
namespace JUNAID_MALIK
{
    // The Main Function of namespace is ignore the SPACE
    class Program  //Class its Stor the Multiple objects
    {
        //Main function (declair or Predefault start from
Main function
        static void Main(string[] args)
        {
            //Declaration in Variable
            int myNum = 10;
            double mydouble = 2.98;
            //{0} use for show the output on Screen
            //if we want in single Line we use index like
as {0},{1}.{2}
            Console.WriteLine("The Value of INT is : {0}",
myNum);
            Console.WriteLine("The Value of INT is : {0}
and The Value of Double is : {1}", myNum, mydouble);
            Console.ReadLine();
        }
    }
}
```

Here is OUT PUT:

```
The Value of INT is : 10

The Value of INT is : 10 and The Value of Double is :
2.98
```

TAKING INPUT FROM USER:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace JUNAID_MALIK
{
    class Program
    {
            static void Main(string[] args)
        {
            string name;
            Console.WriteLine("Enter Your Name :");
            name = Console.ReadLine();
            Console.WriteLine("Your Name is :{0}",name);
            Console.ReadLine();
        }
    }
}
```

OUT PUT

```
Enter Your Name :

JUNAID

Your Name is :JUNAID
```

AL-JUNAID INSTITUTE OF GROUP

Athematic OPERATIONS:

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

namespace JUNAID_MALIK
{
    internal class Program
    {
        static void Main(string[] args)
        {
            int num1 = 4;
            int num2 = 2;
            int ttlnum = num1 + num2;
            int ttlnum1 = num1 / num2;
            Console.WriteLine("The SUM of Two Int
is : {0}",ttlnum);
            Console.WriteLine("The DIVISION of Two
Int is : {0}", ttlnum1);
            Console.ReadLine();
        }
    }
}
```

STRING FUNCTION:

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

```
namespace ConsoleApp114
{
    internal class Program
    {
        static void Main(string[] args)
        {
            string mystring = "JUNAID
MALIK";
            string mystring1 = "My age is
21";
            //You can alos Concatigate
            string mystr =
string.Concat(mystring, " ", mystring1);
            Console.WriteLine("The Length
of My STR is :{0}", mystring.Length);
            Console.WriteLine("LOWER case
is : {0}", mystring.ToLower());
            Console.WriteLine("Cancat
String : {0}", mystr);
            Console.ReadLine();
        }
    }
}
```

OUT PUT:

```
The Length of My STR is :12

LOWER case is : junaid malik

Cancat String : JUNAID MALIK My age is 21
```

Mathematical Function:

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

namespace ConsoleApp114
{
    internal class Program
    {
        static void Main(string[] args)
        {
            double var1 = Math.Sqrt(36);
            double var2 = Math.Max(1024,
1022);
            double var3 = Math.Round(9.99);
            double var4 = Math.Abs(-40.7);
            Console.WriteLine("Square Root
of 36 is : {0}",var1);
            Console.WriteLine("MAX of VAR2
: {0}", var2);
            Console.WriteLine("RoundOFF :
{0}", var3);
            Console.WriteLine("absolute
(positive) value : {0}", var4);
            Console.ReadLine();
        }
    }
}
OUT PUT:
Square Root of 36 is : 6
MAX of VAR2 : 1024
RoundOFF : 10
absolute (positive) value : 40.7
```

LOOPS in C#:

- While Loop
- For Loop
- Foreach Loop

WHILE LOOP:

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

namespace ConsoleApp114
{
    internal class Program
    {
        static void Main(string[] args)
        {
            int x = 0;
            while (x < 10)
            {
                Console.WriteLine(x);
                x++;
            }
        }
    }
}
```

Do While Loop:

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

namespace ConsoleApp114
{
    internal class Program
    {
        static void Main(string[] args)
        {
            int i = 0;
            do
            {
                Console.WriteLine(i);
                i++;
            }
            while (i < 5);
        }
    }
}
```

For Loop:

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

namespace ConsoleApp114
{
    internal class Program
    {
        static void Main(string[] args)
        {
            int table = 9;
            for (int x = 1; x <= 10; x++)
            {
                Console.WriteLine("{0} x {1} ={2}", table, x, table * x);
            }
        }
    }
}
```

OUT PUT:

```
9 x 1 =9

9 x 2 =18

9 x 3 =27

9 x 4 =36

9 x 5 =45

9 x 6 =54

9 x 7 =63

9 x 8 =72

9 x 9 =81

9 x 10 =90
```

ForEach Loop:

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

namespace ConsoleApp114
{
    internal class Program
    {
        static void Main(string[] args)
        {
            string[] sub = { "CS", "MTH", "MGT",
"STA", "ENG" };
            foreach (string s in sub)
            {
                Console.WriteLine(s);
            }

        }
    }
}
```

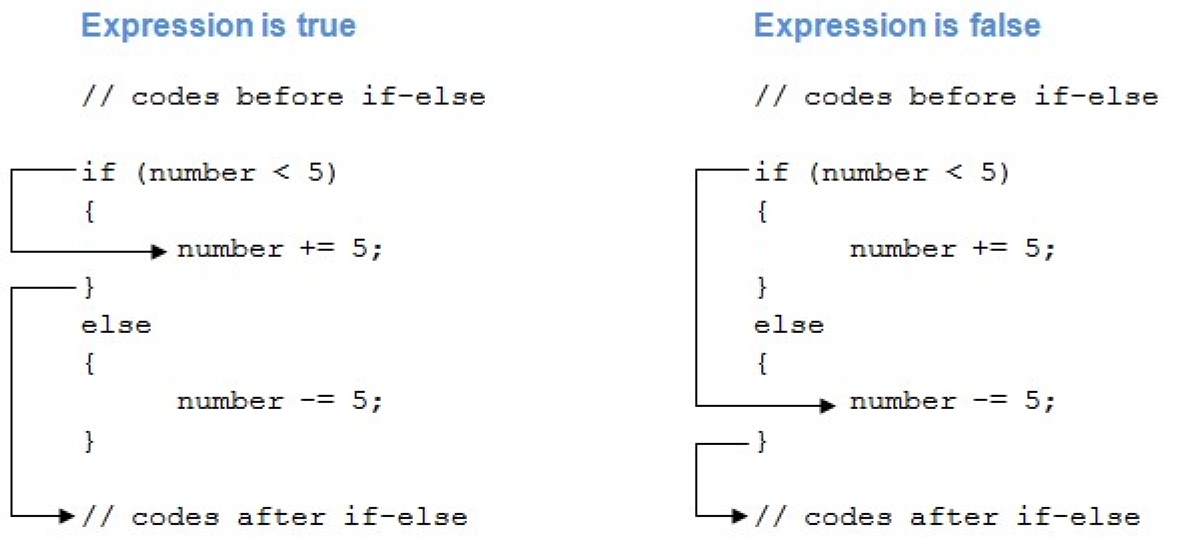
If Else in C#:

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

namespace ConsoleApp114
```

```
{
    internal class Program
    {
        static void Main(string[] args)
        {
            int age = 21;
            if (age < 18)
            {
                Console.WriteLine("You are Eligibe");
            }
            else
            {
                Console.WriteLine("You are not 
Eligibe");
            }
        }
    }
}
```

If else If:



Expression is true

```
// codes before if-else

if (number < 5)
{
        number += 5;
}
else
{
        number -= 5;
}

// codes after if-else
```

Expression is false

```
// codes before if-else

if (number < 5)
{
        number += 5;
}
else
{
        number -= 5;
}

// codes after if-else
```

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

namespace JUNAID_MALIK
{
    internal class Program
    {
        static void Main(string[] args)
        {
            int age = 21;
            if (age > 21)
            {
                Console.WriteLine("You are Eligibe");
            }
            else if (age<18)
            {
                Console.WriteLine("You are not
Eligibe");
            }
            else
                {
                Console.WriteLine("PLZ Enter Valid
age");
            }

        }
    }
}
```

USER INPUT:

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

namespace ConsoleApp115
{
    internal class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("JUNAID
MALIK");
            Console.Write("PLZ Chose:  1, 2,
or 3: ");
            string userChoice =
Console.ReadLine();
            string message = "";

            if (userChoice == "1")
                message = "You are Eligible
in Party!";
            else if (userChoice == "2")
                message = "You are not
Eligible In Party!";
            else if (userChoice == "3")
                message = "You are restigate
from UNI!";
            else
                message = "Better luck next
time.";

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

OUT PUT:

JUNAID MALIK

PLZ Chose: 1, 2, or 3: 1

You are Eligible in Party!

SWITCH STATEMENT:

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

namespace ConsoleApp116
{
    internal class Program
    {
        static void Main(string[] args)
        {
            int day;
            Console.WriteLine("Enter Your
Choice");
            day =
Convert.ToInt32(Console.ReadLine());

            switch (day)
            {
                case 1:
                    Console.WriteLine("Monday");
                    break;
                case 2:

Console.WriteLine("Tuesday");
                    break;
                case 3:

Console.WriteLine("Wednesday");
                    break;
                case 4:

Console.WriteLine("Thausrday");
                    break;
                case 5:
                    Console.WriteLine("Friday");
                    break;
                case 6:

Console.WriteLine("Saturday");
                    break;
                case 7:
                    Console.WriteLine("Sunday");
                    break;
                default:
                    Console.WriteLine("PLZ VALID
Choice");
                         break;
            }

        }
    }
}
```

AL-JUNAID INSTITUTE OF GROUP

Break and Continue Statement in C#:

Break Statement:

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

namespace ConsoleApp116
{
    internal class Program
    {
        static void Main(string[] args)
        {
            for (int i = 0; i < 10; i++)
            {
                if (i == 4)
                {
                    break;
                }
                Console.WriteLine(i);
            }

        }
    }
}
```

Continue Statement:

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

namespace ConsoleApp116
{
    internal class Program
    {
        static void Main(string[] args)
        {
            for (int i = 0; i < 10; i++)
            {
                if (i == 4)
                {
                    continue;
                }
                Console.WriteLine(i);
            }

        }
    }
}
```

Break and Continue Statement in LOOP:

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

namespace ConsoleApp116
{
    internal class Program
    {
        static void Main(string[] args)
        {
            int i = 0;
            while (i < 10)
            {
                Console.WriteLine(i);
                i++;
                if (i == 4)
                {
                    break; or continue;

                }
            }

        }
    }
}
```

FUNCTION CALL

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

namespace ConsoleApp116
{
    internal class Program
    {
        static void funct()
        {
            Console.WriteLine("Function CAll");
            Console.ReadLine();
        }
        static void Main(string[] args)
        {
            funct();
        }
    }
}
```

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

namespace ConsoleApp116
{
    internal class Program
    {
        static void funct()
        {
            int a = 5;
            int b = 4;
            int c = a + b;
            Console.WriteLine("The sum of two
Integer is: {0}", c);
            Console.ReadLine();
        }
        static void Main(string[] args)
        {
            funct();
        }
    }
}
```

AL-JUNAID INSTITUTE OF GROUP

C# : What is OOP?

- OOP stand for Object-Oriented Programming
- OOP is Faster and Easier to execute
- OOP provides a clear Structure
- OOP helps to keep the C# code DRY "don't Repeat Yourself", and makes the code easier to maintain,modify and debug.
- OOP makes it possible to create full reusable Applications with CODE shorter developer time.

OOP CONCEPT:

- CLASS
- OBJECT
- POLYMORPHISM (The Method in which we manage the OBJECT)
- INHERITENCE (Parent class ki property Lana)
- ENCAPSULATION (BUNDLING of DATA. So Much Data convert into single unit)
- ABSTRACTION (Data abstraction. It is ability that hide the Background DATA process)

CLASSES and OBJECT:

- Everything in C# are ASSOCIATED in CLASS and OBJECTS along with attributes and Methods.
- For Example: in Real Life, a car is an object. The car has Attributes, such as weight and color and methods, such as driver and Bike
- A class is like an object Constructor or a "BLUEPRIN" for creating objecs.

Class	OBJECTS
SUBJECTS	CSS,MTH,MGT

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

namespace ConsoleApp116
{
    class sub
    {
        string a = "CS";
        int b = 89;
        string c = "A+";

        static void Main(string[] args)
        {
            sub myobj = new sub();
            Console.WriteLine(myobj.a);
            Console.WriteLine(myobj.b);
            Console.WriteLine(myobj.c);
            Console.ReadLine();
        }
    }
}
```

OUT PUT:

```
CS

89

A+
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace MALIK_JUNAID
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object
sender, EventArgs e)
        {

        }

        private void
textBox2_TextChanged(object sender, EventArgs
e)
        {

        }
        private void button1_Click(object sender, EventArgs e)
        {
            string email = textBox1.Text;
            string pass = textBox1.Text;
            if((this.textBox1.Text=="junaidfazal08@gmail.com") && (this.textBox2.Text=="junaid123"))
            {
                MessageBox.Show("Congratulation you are successfully Log in");
            }
            else
            {
                MessageBox.Show("PLZ try again Later");
            }
        }

        private void button2_Click(object sender, EventArgs e)
        {
            this.Close();
        }
    }
}
```

Create an Object

An object is created from a class. We have already created the class named `Person`, so now we can use this to create objects.

CODE:

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

namespace junaid_malik
{
    class Person
    {
        string person_name = "JUNAID MALIK";

        static void Main(string[] args)
        {
            Person myObj = new Person();

Console.WriteLine(myObj.person_name);
        }
    }
}
```

Class Members

In the previous chapter, you learned that variables inside a class are called fields, and that you can access them by creating an object of the class, and by using the dot syntax (.).

The following example will create an object of the Car class, with the name myObj. Then we print the value of the fields color and maxSpeed:

CODE:

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

namespace junaid_malik
{
    class Car
    {
        string color = "red";
        int maxSpeed = 200;

        static void Main(string[] args)
        {
            Car myObj = new Car();
            Console.WriteLine(myObj.color);
            Console.WriteLine(myObj.maxSpeed);
        }
    }
}
```

C# Constructors

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

namespace junaid_malik
{
    // Create a PERSON class
    class person
    {
        public string person_quality;  // Create a field

        // Create a class constructor for the person class
        public person()
        {
            person_quality = "MALE"; // Set the initial
value for For Person Quality
        }

        static void Main(string[] args)
        {
            person Ford = new person();  // Create an
object of the Person Class (this will call the constructor)
            Console.WriteLine(Ford.person_quality);  //
Print the value of Person Quality
        }
    }
}
```

OUTPUT:

MALE

C# Properties (Get and Set)

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

namespace junaid_malik
{
    class Person
    {
        private string name; // field
        public string Name   // property
        {
            get { return name; }
            set { name = value; }
        }
    }

    class Program
    {
        static void Main(string[] args)
        {
            Person myObj = new Person();
            myObj.Name = "JUNAID MALIK";
            Console.WriteLine(myObj.Name);
        }
    }
}
```

OUTPUT:

```
JUNAID MALIK
```

C# Inheritance

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

namespace junaid_malik
{
    class Vehicle  // base class (parent)
    {
        public string brand = "Ford";  // Vehicle field
        public void honk()             // Vehicle method
        {
            Console.WriteLine("Tuut, tuut!");
        }
    }

    class Car : Vehicle  // derived class (child)
    {
        public string modelName = "Mustang";  // Car field
    }

    class Program
    {
        static void Main(string[] args)
        {
            // Create a myCar object
            Car myCar = new Car();

            // Call the honk() method (From the Vehicle class)
on the myCar object
            myCar.honk();

            // Display the value of the brand field (from the
Vehicle class) and the value of the modelName from the Car
class
            Console.WriteLine(myCar.brand + " " +
myCar.modelName);
        }
    }}
```

C# Polymorphism

Polymorphism and Overriding Methods

Polymorphism means "many forms", and it occurs when we have many classes that are related to each other by inheritance.

Like we specified in the previous chapter; **Inheritance** lets us inherit fields and methods from another class. **Polymorphism** uses those methods to perform different tasks. This allows us to perform a single action in different ways.

For example, think of a base class called Animal that has a method called animalSound(). Derived classes of Animals could be Pigs, Cats, Dogs, Birds - And they also have their own implementation of an animal sound (the pig oinks, and the cat meows, etc

```
    class Animal  // Base class (parent)
    {
        public void animalSound()
        {
            Console.WriteLine("The animal makes a
sound");
        }
    }

    class Pig : Animal  // Derived class (child)
    {
        public void animalSound()
        {
            Console.WriteLine("The pig says: wee wee");
        }
    }

    class Dog : Animal  // Derived class (child)
    {
        public void animalSound()
        {
            Console.WriteLine("The dog says: bow wow");
        }
    }
```

2nd CODE:

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

namespace junaid_malik
{
    class Animal
    {
        public int rent_of_anmal;
        public Animal(int rentOfAnimal)

        {
            this.rent_of_anmal = rentOfAnimal;
        }
        public virtual int price_calculation(int price)
        {
            return rent_of_anmal + price;
        }


        class horse : Animal
        {
            public horse(int rent_of_animal) : base(rent_of_animal) { }
            public override int price_calculation(int price)
            {
                Console.WriteLine("calculating the price of horse");
                return base.price_calculation(price);
            }
            class elephent : Animal
            {
                public elephent(int rent_of_animal) : base(rent_of_animal) { }
                public override int price_calculation(int price)
                {
                    Console.WriteLine("calculating the price of elephent");
                    return base.price_calculation(price);
                }
                class camel : Animal
                {
                    public camel(int rent_of_animal) : base(rent_of_animal) { }
                    public override int price_calculation(int price)
                    {
                        Console.WriteLine("claculating the price of camel");
                        return base.price_calculation(price);
                    }

                    static void Main(string[] args)
                    {
                        Console.WriteLine("please enter the last digit of your vu id
as number of animal you want to buy");
                        int noOfAnimal = int.Parse(Console.ReadLine());
                        Console.WriteLine("please enter the first 5 digit of your vu
id as rent of animal");
                        int rent = int.Parse(Console.ReadLine());
                        int total_price = 0;
                        for (int i = 0; i <= noOfAnimal; i++)
                        {
                            Console.WriteLine("if you want to buy horse, please
press 1 \n" +
                              "if you want to buy elephent, please press 2 \n" +
                            "if you want to buy camel, please press 3 \n" +
                            "or if you want to exit press 0");
                            int choice = int.Parse(Console.ReadLine());
                            if (choice == 0)
                            {
                                break;
                            }
                            if (choice == 1)
                            {
                                Animal horse = new horse(rent);
                                int price = horse.price_calculation(50000);
                                Console.WriteLine("price for single animla you just
bought is:" + price);
                                total_price += price;
                                Console.WriteLine("Total price for all the animal's
you have bought till now :" + total_price);
                            }
                            if (choice == 2)
                            {
                                Animal elephent = new elephent(rent);
                                int price = elephent.price_calculation(80000);
                                Console.WriteLine("price for single animla you just
bought is:" + price);
                                total_price += price;
                                Console.WriteLine("Total price for all the animal's
you have bought till now" + total_price);
                            }
                            if (choice == 3)
                            {
                                Animal camel = new camel(rent);
                                int price = camel.price_calculation(100000);
                                Console.WriteLine("price for single animla you just
bought is:" + price);
                                total_price += price;
                                Console.WriteLine("Total price for all the animal's
you have bought till now" + total_price);
                            }
                        }

                    }
                }
            }
        }
    }
}
```

C# Abstraction

Abstract Classes and Methods

Data **abstraction** is the process of hiding certain details and showing only essential information to the user.
Abstraction can be achieved with either **abstract classes** or **interfaces** (which you will learn more about in the next chapter).

The abstract keyword is used for classes and methods:

- **Abstract class:** is a restricted class that cannot be used to create objects (to access it, it must be inherited from another class).
- **Abstract method:** can only be used in an abstract class, and it does not have a body. The body is provided by the derived class (inherited from).

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

namespace junaid_malik
{
    // Abstract class
    abstract class Animal
    {
        // Abstract method (does not have a body)
        public abstract void animalSound();
        // Regular method
        public void sleep()
        {
            Console.WriteLine("BUFFELOW");
        }
    }

    // Derived class (inherit from Animal)
    class Pig : Animal
    {
        public override void animalSound()
        {
            // The body of animalSound() is provided
here
            Console.WriteLine("The pig says: COW");
        }
    }

    class Program
    {
        static void Main(string[] args)
        {
            Pig myPig = new Pig();  // Create a Pig
object
            myPig.animalSound();
            myPig.sleep();
        }
    }
}
```

C# Interface

To access the interface methods, the interface must be "implemented" (kinda like inherited) by another class. To implement an interface, use the : symbol (just like with inheritance). The body of the interface method is provided by the "implement" class. Note that you do not have to use the override keyword when implementing an interface

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

```
namespace junaid_malik
{
    // Interface
    interface IAnimal
    {
        void animalSound(); // interface method
(does not have a body)
    }

    // Pig "implements" the IAnimal interface
    class Pig : IAnimal
    {
        public void animalSound()
        {
            // The body of animalSound() is
provided here
            Console.WriteLine("The pig says:
BUFFELLOW");
        }
    }

    class Program
    {
        static void Main(string[] args)
        {
            Pig myPig = new Pig();  // Create a
Pig object
            myPig.animalSound();
        }
    }
}
```

C# Enum

An enum is a special "class" that represents a group of **constants** (unchangeable/read-only variables).

To create an enum, use the enum keyword (instead of class or interface), and separate the enum items with a comma

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

namespace junaid_malik
{
    class Program
    {
        enum john
        {
            JUNAID,
            MALIK,
            MULTAN
        }
        static void Main(string[] args)
        {
            john myVar = john.JUNAID;
            Console.WriteLine(myVar);
        }
    }
}
```