9. Operators :-

Arithmetic operators (+, −, *, /, %, ++, −−)

Relational operators (==, !=, <, >, <=, >=)

Logical operators (true, false)

- && → and
- || → or

Assignment operator (+=, −=, *=, /=, %=, ^=)

Ex:- a^=2 means a = a^2

Double x = 5;
Double a = Math.Pow(x, 3);
// x^3 a = x^3
= 5^3 = 125

10. COMMENTS :-

// This is a Single line comment.

/* This is a
multi line
comments */

11. Region :-

#region
// block of code
#region

> // this can hide a block of code under it

12. Name conventions :-

It's types are →
- Pascal casing (Ex:- int MyFirstNo)
- camal casing (Ex:- string phoneNo)

13. Break point :-

- Syntax errors
- Runtime errors
- Logic errors

14. Keyboard Shortcut :-

Math class

```
using system;
namespace myfirstProgram {
    class Program {
        Public Static void Main (String[] args){
            double x = 3;
            double a = Math.Pow(x, 3);
                // a = 3^3
            Console.Writeline(a);
            Console.Readkey();
            }
        }
    }
```

By using Math class, we use Math class attributes

- pow( )
- PI ( )
- Max( )
- Min ( )
- Round( )
- Sin ( )
- Sqrt-( )

Ex:- $\sqrt{a^2+b^2}$

if $a = 2$, $b = 3$ | then $a^2 = 4$, $b^2 = 9$ | then $\sqrt{4+9}$ $\cdot$ $\sqrt{13}$ = 3.605 ...

Main code

```
double a2 = convert.Todouble (Console.Readline());
double b2 = Convert.ToDouble(console.Readline());
double a2 = Math.Pow(a2, 2);
       b2 = Math.Pow(b2, 2);
double c  = Math.Sqrt(a2 + b2);
```

(*) (*) from youtube video

Note 'int' input by the user →

```
Console.WriteLine("What's your age.?");
int age= Convert.ToInt32(Console.ReadLine());
```

```
Console.WriteLine("What's your hight ?");
double Hight = Convert.ToDouble(Console.ReadLine());
```

(*) (*) next() for int value

```
Random random = new Random();
int num = random.next(1,21);
Console.WriteLine(num);
```

nextdouble() for double value

```
Random random = new Random();
double num = random.nextDouble(11, 23);
Console.WriteLine(num);
```

Ex:- $\sqrt{a^2 + b^2}$ Code this —

(*) String Methods :=

```
String name = "kuntal";
Console.WriteLine(name.ToUpper());
//KUNTAL
```

- .ToLower();
- .Replace(old, new);
- .Insert(index, new);
- .Length
- .Substring();

String interpolation

```
string FirstName = "Bro";
string LastName = "code";
int Age = 21;
Console.WriteLine($"Hello {FirstName} {LastName}.");
Console.WriteLine($"Your age is: {age}.");
```

15. Condition controls :-

If else Statement >

```
if (//condition)
{
}
else
{
}
```

```
if (  )
{
}
elseif (  )
{
}
else {
}
```

Switch Statement >

```
Switch (//condition)
{
case ( ):
        //Statement
        break;
default :
        //Statement
        break;
```

Ternary Statement > `//condition ? //Statement1 : // Statement2`

14. Loops :-

- Entry control loop (for, while loop)
- Exit control loop (do-while)

while loop >

```
while (condition)
{
    //loop Statement
}
```

do-while >

```
do
{
    //Statement
} while (condition)
```

for loop >

```
for (int i = 0; i<=10; i++)
{
    //Statement
}
```

Loops ⇒

for loop syntax

```
for( initialization ; condition ; incr/decr)
    {
    // code to be executed
    }
```

while loop syntax

```
initialization ;
while ( condition)
    {
    // code to be executed
    }
incr/decr ;
```

do-while loop syntax

```
initialization ;
do {
    // code to be executed
    incr/decr ;
} while (condition) ;
```

Break statement ⇒

Break is used to break loop. It break the current flow of the program at that given condition. In case of inner loop it breaks only inner loop.

```
....
....
else
{
// code to be executed if all
the conditions falsh
}
```

Switch →

Switch syntax

```
switch (expression) {
    case value1:
        // code to be executed
        break;
    case value2:
        // code to be executed
        break;
    ....
    ....
    default :
        // code to be execute
        // if all case are not
            match
        break;
}
```

Continue →

It is used to continue loop. It continues the current flow of the program and Skip the remaining code at specified condition In case of inner loop, it continues only inner loop.

Goto Statement →

It is used to transfer control to the other part of the program. It unconditionally jumps to the specified level.

```
using system;
namespace consoleAppliation{
        public class Program{
        public static void Main(string[] args)
        {
here:
console.Writeline("hi,");
console.writeline("good morning");
goto here;
}
}
}
```

System

```
hi,
good morning
```

Foreach >. | foreach (var item in list or array)
{ :
// Statement
}

5. Jump statement

↳ | break ✓
continue ✓
goto
return
throw

Note ⇒

project on c # → see it

Other project ⇒ Standard calculator
⇒ Billing application.
⇒ weather app. (by using weather Api)

using c# & Dot_Net with SQL

Ex:- '*' pattern printing patterns by using loops.

Ex:- 1. Number guessing game
2. Rock - Paper - scissors.
3. calculator program. (by using switch)

Random random = new Random();

Time stamp 1:46

Array (1-D Array)

A variable that can store multiple values. fixed size

syntax

```
string [] cars = {"BMW", "Mustang", "Corvette"};
Console.WriteLine(cars[0]); // BMW
cars[0] = "Teshla";
Console.WriteLine(cars[0]); // Teshla.
```

Array declaration process

```
for(int i=0; i< cars.length; i++)
{
    Console.WriteLine(cars[i]);
}
```

// getting elements from array

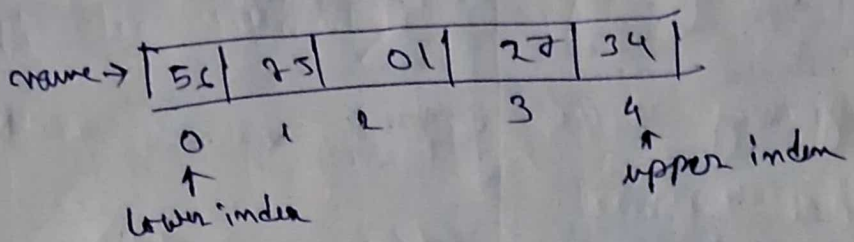
output

```
// Teshla
// Mustang
// corvette
```

syntax

```
string [] cars = new string [3];
cars[0] = "Teshla";
cars[1] = "Mustang";
cars[2] = "corvette";
Console.WriteLine(cars[1]); // Mustang
```

Array (1D)



datatype [] name; | Syntax.

declaration of array.

```
int [] number;
number = new int [4] {3, 4, 5, 6};
```

```
string [] days;
days = new string [] { "sunday", "Monday", "Tuesday"
                       "wednesday", "Thursday", "Friday" };

foreach (string item in days)
{
    console.WriteLine (item);
}
```

output

```
sunday
Monday
Tuesday
wednesday
Thursday
Friday
Sat
```

- **foreach loop** = a simpler way to iterate over an array, but- It's less flexible.

Foreach loop syntax.

```
String [] cars = {"BMW", "Mustang", "corvette"};

foreach (string car in cars)
{
    console.Writeline(car)
}
```

- **Methods or Function** → function call, function ~~declination~~ declaration.

```
.... Main(string[] args) {
    String name = "kuntal";
    int age = 24;
    singGoodMorning(name, age);
    console.Readkey();
}
```

→ function call

```
Static void singGoodMorning (string name, int old)
{
    console.Writeline("Hello, Good Morning," name);
    console.Writeline("your age is: ", Age);
}
```

→ function declaration

- **Return keyword** :-

```
... Main(string[] args) {
    double a, b, result;
    ~~TotalResult(a, b);~~
    a = Convert.Todouble(console.Readline());
    b = "  "  "  ("  "  ());
    Writeline("result-="; result);
Static ~~void~~ double ~~result~~ TotalR (double x, double y){
    double z = x + y; return z; }
```

console.Writeline("Enter a=");
" " ("Enter b=");
result= TotalR(a,b);

- Method overload (we can pass 3-Argument in a function)

```
... Main(string[] args) {
        double total;
        total = Multiply(2, 3, 4);
        Console.WriteLine(total);
        Console.Readkey();
    }
Static double Multiply(double a, double b, double c)
    {
        return a*b*c;
    }
```

- Params keywords :-

```
..... Main (string[] args) {
    double total = checkout(3.99, 5.75, 15, 1.00, 10.25);
        Console.WriteLine(total);
        Console.Readkey();
    }
~~public~~ static double checkout(params double[] prices)
                                    // prices array
    {
        double total = 0;
        foreach (double price in prices)
        {
            total = total + price;
        }
        return total;
    }
```

// Params is a method parameter that's take a variable number of argument.

The parameter types must be a single-dimensional (1D) array.

Time stamp 2:08

Billing System (project)

output

Name:-
Sh no:-

1st order item :
1st order item price : _____ // automatic generated

1st " " quantity :

// side total = Item price x quantity = _____ (calculated)

if there is 2nd order chose add or exit
and print then chose exit :- add / exit

continue.

total bill to be paid :- _____
// autogenerated

Done

Try & catch

```
try {
    Console.WriteLine("Enter a number:");
    double x = Convert.ToDouble(Console.ReadLine());
    Console.WriteLine("Enter another number:");
    double y = Convert.ToDouble(Console.ReadLine());
    double result = x/y;
    Console.WriteLine("The result is:", result);
}
catch (FormatException e) {
    Console.WriteLine("Enter only the number");
}
```

```
catch ( DivideByZeroException e) {
        Console.WriteLine("You can't divide by zero");
}
finally {
        Console.WriteLine("Thank you");
}
Console.ReadKey();
}
```

conditional operator / Ternary operator

```
variable = (condition) ? statement 1 : statement 2;
```

Multi-D-Array (2-D Array)

```
String[,] class1 = { {"Kuntal", "Kunal", "Keka"},
                     {"Riya", "Shreya", "Naman"},
                     {"Rohit", "Sanju", "Mona"}
                   };
```

(0,0 = row, column; 0,1; 0,2; 1,0; 1,1; 1,2; 2,0; 2,1; 2,2)

Pattern that change

```
String[,] class1 [2,1] = "Gita";
for(int i = 0; i < 3; i++) {
    for(int j = 0; j < 3; j++) {
        Console.Write(class1[i,j], " ");
    }
    Console.WriteLine();
}
```

output

Kuntal Kunal Keka
Riya Shreya Naman
Rohit Gita Mona

// this is the change in the place of [2,1] 'Sanju'

String interpolation

```
String FirstName = "Broo";
String LastName = "code";
int Age = 21;
Console.WriteLine($"Hello {FirstName} {LastName}.");
Console.WriteLine($"Your age is: {age}.");
```