**C Sharp**

Terrminal > dotnet new console

**Console**.**WriteLine**("hello world");

Terminal> dotnet run

**Type of data type**

1. **Explicit (clearly defined)**

Int a; - declartion

A=10; - intialization

B=20;

A= ”nazim” not allowed

1. **Implicit**

Var a; - declaration shows arror since don’t know variable which type of data type;

Var a =5; - now after assiging value 5 to variable, a data type is integer automatically.

a=”nazim” - shows error since it is considered integer.

Note- implicit frequently use.

1. **Dynamic;**

Replacement of var;

We can not declare variable with var but can declare with dynamic keywod;

**Dynamic a; - declaration no error;**

Dynamically change its data type

**Dynamic a=5;**

**Dynamic a= “nazim”**

Dynamic slow to application since it is keep changing.

Use cases – when we are not sure which data type is coming and assigning to a varibale.

**example**

int num\_a; //declaration

int num\_b;

num\_a= 5; //initialization

num\_b= 10;

**Console**.**WriteLine**(num\_a+num\_b);//15

**String concatination**

string first\_name = "nazim";

string last\_name = "khan";

string fullname = first\_name+" "+last\_name;

**System**.**Console**.**WriteLine**(fullname);

**string + interger, output is string**

int num\_a; *//declaration*

int num\_b;

num\_a= 5; *//initialization*

num\_b= 10;

**Console**.**WriteLine**("total is " + num\_a+num\_b);//510

//sting + interger, output is string

**Solution to string plus interger**

int num\_a; *//declaration*

int num\_b;

num\_a= 5; *//initialization*

num\_b= 10;

**Console**.**WriteLine**("total is " + (num\_a+num\_b));//15

**Console**.**WriteLine**(("total is " + (num\_a+num\_b)).**GetType**());

**Conversion**

**Note - User input to be considered interger**

int valueOne;

**System**.**Console**.**WriteLine**("please enter value  one");

valueOne = **Convert**.**ToInt32**(**Console**.**ReadLine**());

**System**.**Console**.**WriteLine**(valueOne.**GetType**());// system.Int32

int valueOne;

**System**.**Console**.**WriteLine**("please enter value  one");

valueOne = int.**Parse**(**Console**.**ReadLine**());

**System**.**Console**.**WriteLine**(valueOne.**GetType**()); //sytem.Int32

int num1;

int num2;

**System**.**Console**.**WriteLine**("enter num1");

num1= **Convert**.**ToInt32**(**Console**.**ReadLine**());

**System**.**Console**.**WriteLine**("enter num2");

num2= **Convert**.**ToInt32**(**Console**.**ReadLine**());

**System**.**Console**.**WriteLine**("the result is" +(num1+num2));//22

**IF Else**

**For decision making**

**>, <, = , not = , !=, >=, <=**

int valueOne = 10;

int valueTwo = 20;

if(valueOne>valueTwo){

**System**.**Console**.**WriteLine**("Value One is greater than Value Two");

}

else{

**System**.**Console**.**WriteLine**("value two is greater  than or equal to Value one");

}

**Gate – ||(or), &&(and)**

int valueOne = 10;

int valueTwo = 20;

if(valueOne!=valueTwo && valueOne>valueTwo) {

**System**.**Console**.**WriteLine**("Value One is greater than Value Two");

}

else{

**System**.**Console**.**WriteLine**("value two is greater  than or equal to Value one");

}

//false

Both condition should be true for true

int valueOne = 10;

int valueTwo = 20;

if(valueOne!=valueTwo || valueOne>valueTwo) {

**System**.**Console**.**WriteLine**("Value One is greater than Value Two");

}

else{

**System**.**Console**.**WriteLine**("value two is greater  than or equal to Value one");

}

//true

At least one condition should be true

**Else if**

int num = -10;

if (num > 0)

{

**Console**.**WriteLine**("Number is positive");

}

else if (num < 0)

{

**Console**.**WriteLine**("Number is negative");

}

else

{

**Console**.**WriteLine**("Number is zero");

}

// Number is negative

Else if (check if block and then i else if block)

int valueOne = 10;

int valueTwo = 20;

if(valueOne>valueTwo){

**System**.**Console**.**WriteLine**("Value One is greater than Value Two");

}

else if(valueOne<valueTwo){

**System**.**Console**.**WriteLine**("vaue  one is lesser than Value two");

}

else if(valueOne!=valueTwo) {

**System**.**Console**.**WriteLine**("value one  is not equal to value two");

}

else{

**System**.**Console**.**WriteLine**("Both values are same");

}

//value one is lesser than value two

Else if (check if block and then i else if block)

**Nested if**

if(true){

    if(true){

**System**.**Console**.**WriteLine**("first true");

    }

    if(true){

**System**.**Console**.**WriteLine**("second true");

    }

}

else{

**System**.**Console**.**WriteLine**("none of the above are true");

}

//first true

//secont true

if(false){

    if(true){

**System**.**Console**.**WriteLine**("first true");

    }

    if(true){

**System**.**Console**.**WriteLine**("second true");

    }

}

else{

**System**.**Console**.**WriteLine**("none of the above are true");

}

// none of the above are true

**If without else**

if(true)

**System**.**Console**.**WriteLine**("this is first line");

**System**.**Console**.**WriteLine**("this is second line");

//this is first line

//this is second line

if(false)

**System**.**Console**.**WriteLine**("this is first line");

**System**.**Console**.**WriteLine**("this is second line");

//this is second line

if(true){

**System**.**Console**.**WriteLine**("first true");

}

if(true){

**System**.**Console**.**WriteLine**("second true");

}

if(false){

**System**.**Console**.**WriteLine**("third true");

}

else{

**System**.**Console**.**WriteLine**("third false");

}

If body block

**Switch case (alternate of else if)**

int day =5;

switch(day){

    case 1:

**System**.**Console**.**WriteLine**("Monday");

    break;

    case 2:

**System**.**Console**.**WriteLine**("Tuesday");

    break;

    case 3:

**System**.**Console**.**WriteLine**("Wednesday");

    break;

    case 4:

**System**.**Console**.**WriteLine**("Thursday");

    break;

    case 5:

**System**.**Console**.**WriteLine**("Friday");

    break;

    case 6:

**System**.**Console**.**WriteLine**("Saturday");

    break;

    case 7:

**System**.**Console**.**WriteLine**("Sunday");

    break;

default:

**System**.**Console**.**WriteLine**("Invalid day");

break;

}

int valueOne;

int valueTwo;

int choice;

**System**.**Console**.**WriteLine**("Enter the first integer:");

valueOne=int.**Parse**(**Console**.**ReadLine**());

**System**.**Console**.**WriteLine**("Enter the second integer:");

valueTwo=int.**Parse**(**Console**.**ReadLine**());

**System**.**Console**.**WriteLine**("Please Select any number \n 1 for add \n 2 for multi");

choice=int.**Parse**(**Console**.**ReadLine**());

switch(choice){

   case 1:

**System**.**Console**.**WriteLine**("The sum is "+(valueOne + valueTwo));

   break;

   case 2:

**System**.**Console**.**WriteLine**("The multi is "+(valueOne \* valueTwo));

   break;

   default:

**System**.**Console**.**WriteLine**("invalid selection");

   break;

}

**Loop**

for (int i = 1; i<=10; i++)

{

**System**.**Console**.**WriteLine**(2\*i);

}

int i=1;

while(i<=5){

**System**.**Console**.**WriteLine**(2\*i);

    i++;

}

int valueOne;

int valueTwo;

int choice=1;

while(choice==1){

**System**.**Console**.**WriteLine**("enter value one");

    valueOne = **Convert**.**ToInt32**(**System**.**Console**.**ReadLine**());

**System**.**Console**.**WriteLine**("enter value two");

    valueTwo= **Convert**.**ToInt32**(**Console**.**ReadLine**());

**System**.**Console**.**WriteLine**("Result is "+(valueOne+valueTwo));

**System**.**Console**.**WriteLine**("Press 1 for continue and  any other number to exit");

    choice = **Convert**.**ToInt32**(**System**.**Console**.**ReadLine**());

**System**.**Console**.**WriteLine**("thank you for using");

}

Program with while

**Array**

*// Syntax: dataType[] arrayName = new dataType[size];*

int[] myArray = new int[5]; *// Creates an integer array with a size of 5*

*// Syntax: dataType[] arrayName = {value1, value2, ..., valueN};*

int[] myArray = { 1, 2, 3, 4, 5 }; *// Creates and initializes an integer array*

int[] myArray = { 1, 2, 3, 4, 5 }; *// Creates and initializes an integer array*

foreach (var item in myArray)

{

**System**.**Console**.**WriteLine**(item);

}

**Array List**

using **System**.**Collections**;

**ArrayList** myArrayList = new **ArrayList**(); *// Creates an empty ArrayList*

*// Adding elements to the ArrayList*

myArrayList.**Add**(10);

myArrayList.**Add**("Hello");

myArrayList.**Add**(3.14);

*// Accessing elements in the ArrayList*

**Console**.**WriteLine**(myArrayList[0]); *// Output: 10*

**Console**.**WriteLine**(myArrayList[1]); *// Output: Hello*

**Console**.**WriteLine**(myArrayList[2]); *// Output: 3.14*

*// Removing elements from the ArrayList*

myArrayList.**Remove**("Hello");

*// Iterating through the ArrayList*

foreach (var item in myArrayList)

{

**Console**.**WriteLine**(item);

}

**Class Object**

**Class**

public class **Student**{

}

Create a file Student.cs -> Class name is Student

**Public, Private, Protected is access modifier**

**Class with variables and methods**

public class **Student**{

    string fristName;

    string lastName;

    string fatherName;

    string gardianName;

    float GPA;

    public void **TakeExam**(){

*//logic to take exam*

    }

    public void **TakeResult**(){

*//logic to take exam*

    }

}

File name Student.cs

**Class with property and methods**

public class **Student**{

    public int Id{get; set;}

    public string Name{get; set;}

    public string fatherName{get; set;}

    public int rollNumber{get; set;}

    public void **saveStudent**(){

**System**.**Console**.**WriteLine**("Saving the student");

    }

    public void **showFullName**(){

**System**.**Console**.**WriteLine**("student full name");

    }

}

File name Student.cs

**Creating object of class Student**

**Student** Nazim = new **Student**();

Nazim.Id=1;

Nazim.Name="nazim Khan";

Nazim.fatherName="asgar ali khan";

Nazim.rollNumber=001;

Nazim.**saveStudent**();

Nazim.**showFullName**();

File name Program.cs

**Class with property and methods with perameter and agrument**

public class **Student**{

    public int Id{get; set;}

    public string Name{get; set;}

    public string fatherName{get; set;}

    public int rollNumber{get; set;}

    public void **saveStudent**(string name){

**System**.**Console**.**WriteLine**(name +"Saving the student");

    }

    public void **showFullName**(){

**System**.**Console**.**WriteLine**("student full name");

    }

}

class

**Student** Nazim = new **Student**();

Nazim.Id=1;

Nazim.Name="nazim Khan";

Nazim.fatherName="asgar ali khan";

Nazim.rollNumber=001;

Nazim.**saveStudent**(Nazim.Name);//pass argument to method

Nazim.**showFullName**();

object

**passing whole object**

public class **Student**{

    public int Id{get; set;}

    public string Name{get; set;}

    public string fatherName{get; set;}

    public int rollNumber{get; set;}

    public void **saveStudent**(string name){

**System**.**Console**.**WriteLine**(name +"Saving the student");

    }

    public void **showFullName**(**Student** student){

**System**.**Console**.**WriteLine**(student.Name + " "+ student.fatherName);

    }

}

**Student(class) student(object)**

**Student** Nazim = new **Student**();

Nazim.Id=1;

Nazim.Name="nazim Khan";

Nazim.fatherName="Asgar ali khan";

Nazim.rollNumber=001;

Nazim.**saveStudent**(Nazim.Name);

Nazim.**showFullName**(Nazim);

**Passing whole object to method**

**Class method with return**

**Note void not works with return keyword, replace it with data type of return**

public class **Student**{

    public int Id{get; set;}

    public string Name{get; set;}

    public string fatherName{get; set;}

    public int rollNumber{get; set;}

    public void **saveStudent**(string name){

**System**.**Console**.**WriteLine**(name +"Saving the student");

    }

    public string **showFullName**(**Student** student){

        return student.Name + " "+ student.fatherName;

    }

}

**class**

**object receiving the value retuned by method**

**Student** Nazim = new **Student**();

Nazim.Id=1;

Nazim.Name="nazim Khan";

Nazim.fatherName="Asgar ali khan";

Nazim.rollNumber=001;

Nazim.**saveStudent**(Nazim.Name);

string studentDetails = Nazim.**showFullName**(Nazim);

**System**.**Console**.**WriteLine**(studentDetails);

**object**

public class **Student**{

 public int **add**(int num1, int num2){

    return num1+num2;

 }

}

**Calss**

**Student** Nazim = new **Student**();

int add = Nazim.**add**(20,20);

**System**.**Console**.**WriteLine**(add);

**object**

**Method/function overloading concept**

public class **Student**{

 public int **add**(int num1, int num2){

    return num1+num2;

 }

 public int **add**(int num1, int num2, int num3){

    return num1+num2+num3;

 }

}

**Same name method in class with different aguments**

**Student** Nazim = new **Student**();

int add = Nazim.**add**(20,20,20);

**System**.**Console**.**WriteLine**(add);

**Method with 3perameters**

**Inheritance**

**Person class**

public class **Person**{

    public int Id{get; set;}

    public string Name{get; set;}

    public string fatherName{get; set;}

}

**Student Class**

public class **Student** : **Person** {

 public string RollNumber{get; set;}

 public void **SaveStudent**(string name){

**System**.**Console**.**WriteLine**(name+"student save successfully");

 }

}

**Teacher class**

public class **Teacher** : **Person** {

    public string qualification{get; set;}

   public void **SubjectName**(){

**System**.**Console**.**WriteLine**("subject of the Teacher");

   }

}

**Object**

**Student** Nazim =new();

Nazim.Id=03;

Nazim.Name= "nazim khan";

Nazim.fatherName="asgar ali";

Nazim.RollNumber="45";

**System**.**Console**.**WriteLine**(Nazim.Name);

**System**.**Console**.**WriteLine**(Nazim.RollNumber);

var teacher = new **Teacher**();

teacher.Id=5;

teacher.Name="paramjeet";

teacher.fatherName="Mr. Patel";

teacher.qualification="MCA";

**System**.**Console**.**WriteLine**(teacher.qualification);

teacher.**SubjectName**("hind");

**how to create objects of class**

**Student** Nazim = new **Student**();

**Student** Nazim =new();

var teacher = new **Teacher**();

**Function overridding**

**Person class**

public class **Person**{

    public int Id{get; set;}

    public string Name{get; set;}

    public string fatherName{get; set;}

    public void **RandomFunc**(){

**System**.**Console**.**WriteLine**("parent random function");

    }

}

**Student class**

public class **Student** : **Person** {

 public string RollNumber{get; set;}

 public void **SaveStudent**(string name){

**System**.**Console**.**WriteLine**(name+"student save successfully");

 }

 public void **RandomFunc**(){

**System**.**Console**.**WriteLine**("student random function");

 }

}

**Object**

**Student** Nazim =new();

Nazim.**RandomFunc**();

**Note-If inherit class and own class has same function then priority will be given to own class**

**Public, Private and Protected Access Modifier**

**Public**

public class **Student** {

 public string RollNumber{get; set;}

 public void **TestFun**(){

**System**.**Console**.**WriteLine**("hello world");

 }

}

**Class property and method with public access modifier**

**Student** Nazim =new();

Nazim.RollNumber="10";

**System**.**Console**.**WriteLine**(Nazim.RollNumber);

Nazim.**TestFun**();

//10

//hello world

**Object accessing student class public property and method**

**Private**

public class **Student** {

 private string RollNumber{get; set;}

 private void **TestFun**(){

**System**.**Console**.**WriteLine**("hello world");

 }

}

**Class property and method with private access modifier**

**Student** Nazim =new();

Nazim.RollNumber="10";

**System**.**Console**.**WriteLine**(Nazim.RollNumber);

Nazim.**TestFun**();

**Object can not access student class property and method due to private access modifier.**

**Protected**

**Note – Protected property/ method can be used from parent class(Person) to only child class(Student) but can not be access further (like object)**

**Parent Class have proted property**

public class **Person**{

    protected int Id{get; set;}

    public string Name{get; set;}

    public string fatherName{get; set;}

}

**Child Student Class have access parent procted propertty**

public class **Student** : **Person** {

 private string RollNumber{get; set;}

 public void **Mid**(){

    Id=5;

**System**.**Console**.**WriteLine**(Id);

 }

}

**Student class object cant have direct access to parent protected property**

**But indirectly have access**

**Student** Nazim =new();

Nazim.**Mid**();

// Nazim.Id; - error not accessible

**Constructor**

**Same like a method in the class only difference is with same name as class name**

**Constructo is not call it is called automatically**

**Class with constructor**

public class **Student**  {

public string RollNumber{get; set;}

public **Student**(){

**System**.**Console**.**WriteLine**("hello world");

}

}

**Only object**

**Student** Nazim =new();

Output// hello world

**Random**

**cw plus enter or tab = System**.**Console**.**WriteLine**();

**to know the data type**

**System**.**Console**.**WriteLine**(num\_b.**GetType()**);