Visual Studio 2022 :

<https://visualstudio.microsoft.com/downloads/>

Sql Server

<https://www.microsoft.com/en-in/sql-server/sql-server-downloads>

Language : Means of communication

Language has character set, grammer(syntax) , BUT LOGIC REMAINS SAME

Eng > A- Z , noun, pronoun, etc.

Programming language : Means of communication between system and human being

Language has character set, grammer(syntax), BUT LOGIC REMAINS SAME

C , C++ , Fortran , Pascal, COBOL, Visual Basic, Java, python, Javascript, C#

When you are communicating with system , you will write statements (program)

Different approaches are used whenever you are writing a program

1. Procedural Approach
2. Object Oriented Approach

**Using Procedural Approach**

Students comes for enquiry

Counsellor handles enquiry

jhgjgjhgh

Students take admission

Trainer take classes

nbjmhbjmh

Trainer give tests

Trainer check tests

Counsellor give certificate

**Using Object Oriented Approach**

Student > Students comes for enquiry, Students take admission

Trainer > Trainer take classes,Trainer give tests,Trainer check tests

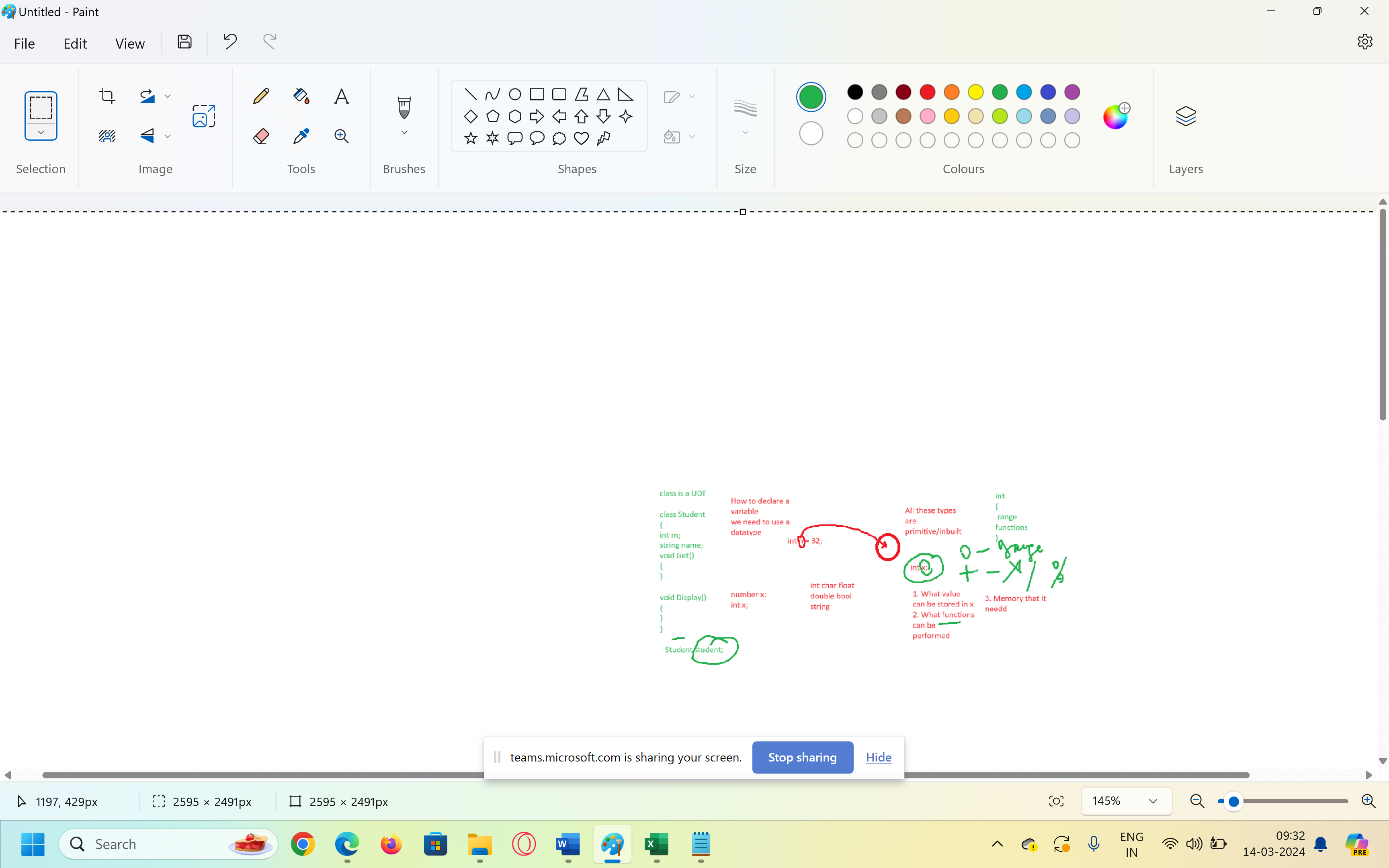
Counsellor > Counsellor handles enquiry, Counsellor give certificate

The system that uses OOA has benefits

1. Are flexible
2. Easy to understand
3. Easy to maintain
4. Suitable for complex systems
5. Easy to test

**Concepts of OOP**

1. Class
2. Object
3. Polymorphism
4. Inheritance
5. Data Abstraction
6. Encapsulation



Variable > When we want to store some value in RAM, We need some location that will hold that value or some time

Int char float Boolean of some type

Inbuilt types

Int, Char, Float, double , bool

Class > Class is a user defined type OR it’s a blueprint based on which multiple objects of same type could be created

Data type : int , char , float

What does a type indicates(What data you can store, what functions we can perform on variable of that type)

Int > what data you can store (numbers)

+ , - \* / % max()

Int x;

Int x;

Int x,y,z,a;

class student

{

Int rn;

String name;

Void get() {}

Void display() {}

}

2.Object > Variable of a class/ Instance of a class

Student student;

Int x;

1. Polymorphism > One name different forms
2. Inheritance > Deriving features of one class into other class
3. Encapsulation > Hiding details which are not needed by the user
4. Abstraction > Showing details which are needed by the user

How do we achieve them > By using Access Specifiers

Private, public, internal ,protected , Internal protected

Any idea about DotNet Framework?

MS DOS > We were to remember the commands to perform basic functions

File Management

Copy source dest

Del filename

Rd dir

Then Win 3.1 came > GUI

Programming lang > C , C++, COBOL , PASCAL, BASIC , FOXPRO

They introduced Visual Effect into the Programming languages

Basic > Visual Basic

Foxpro > Visual Foxpro

C++ > Visual C++

They thought that why don’t we bring these dif languages under one platform > DotNet Platform

Main benefit of DotNet Framework > It supports multiple languages

 it supports more than 60 programming languages such as C#, F#, VB.NET, J#, VC++, JScript.NET, APL, COBOL, Perl, Oberon, ML, Pascal, Eiffel, Smalltalk, Python, Cobra, Ada, etc.

C# , Vb.Net



Framework > Collection of packages (namespaces) , classes

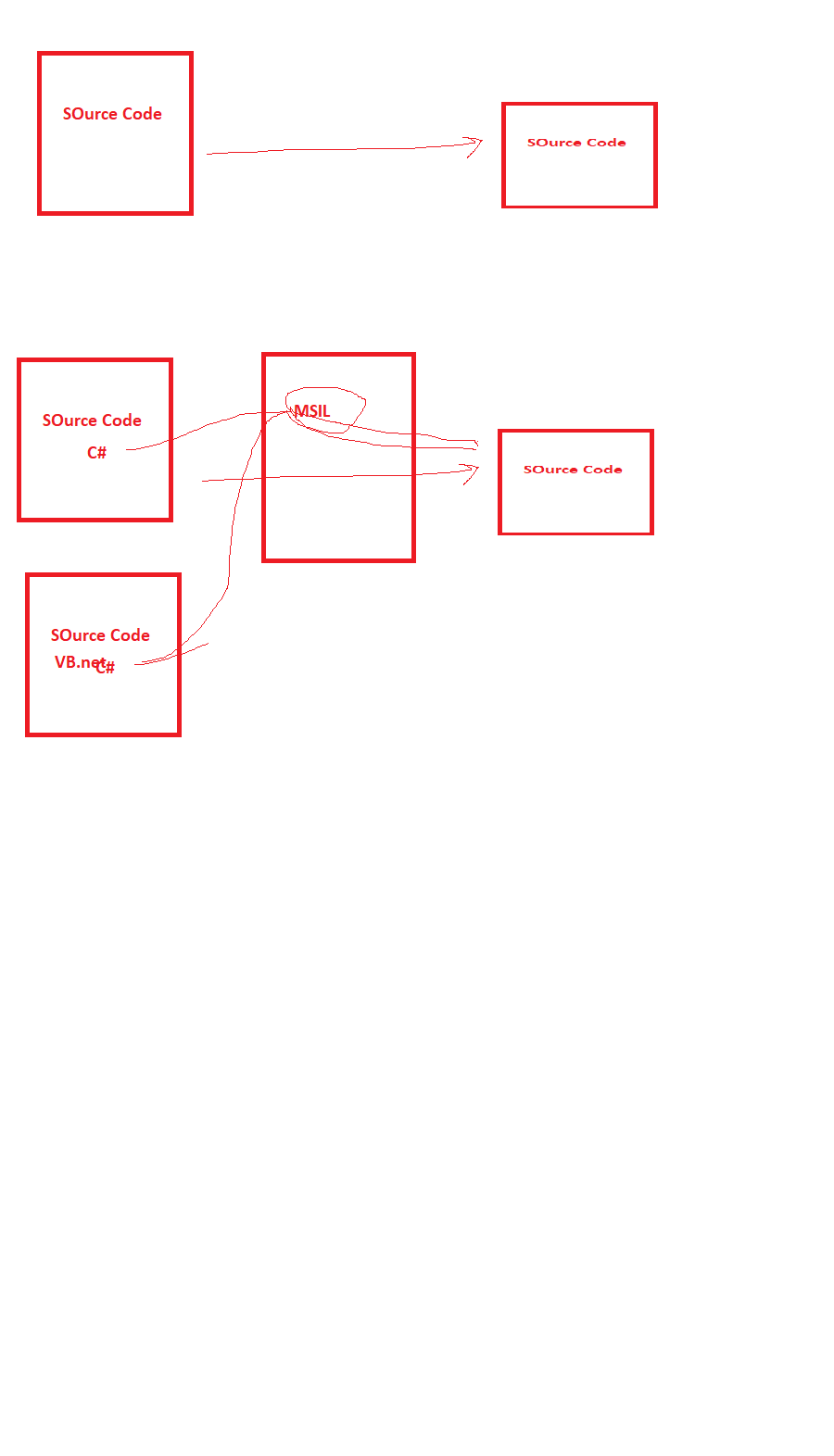


Components of DotNet Framework

1. CLR > Common Language Runtime (It’s a run time environment for DotNet Programs)
2. CTS > Common Type System
3. CLS > Common Language Specification
4. Base Libraries

Compilation > Converting a program from one lang to other lang

C++ , C , Java > Compilation happens on 1 step



C# > Compilation happens on 2 steps

C# > C# compiler will convert program from C# to MSIL, this is a form which is understood by CLR

, after that CLR will compile that program to binary form using JIT compilation

JIT compilation > Complete program is not converted to binary form only, the statements needed at that time are converted to binary form

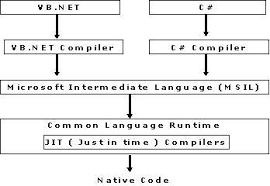
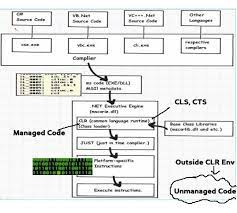
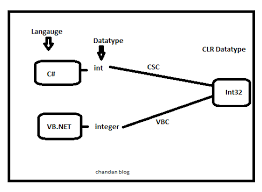
**Functions of CLR**

The heart of DotNet framework

1. Compiles your code
2. Garbage Collection
3. Memory Allocation
4. Type Safety
5. Code Security

CTS > Common Type System

CLS > Common Language Specification

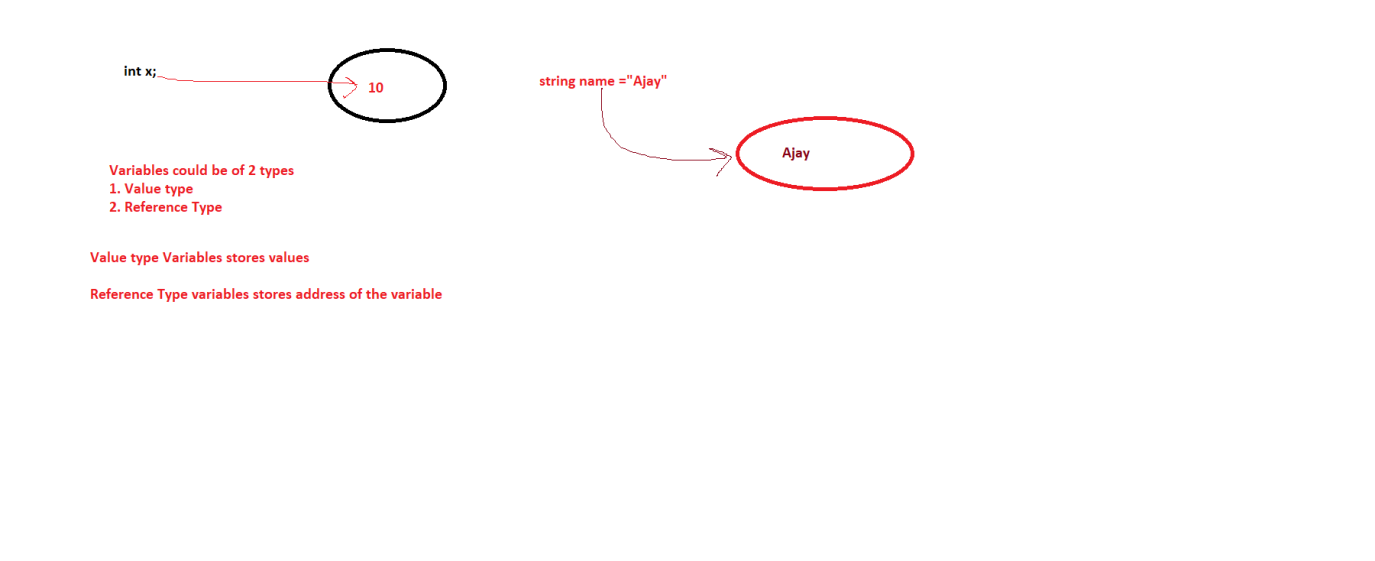


Managed Code > Code which is understood by CLR

UnManaged Code > Code which is not understood by CLR

Variables > Names given to memory locations where the values are stored

Int x



Value Type Variables > int , char , float , bool, enum, struct

Reference Type variables > string, Array, class

