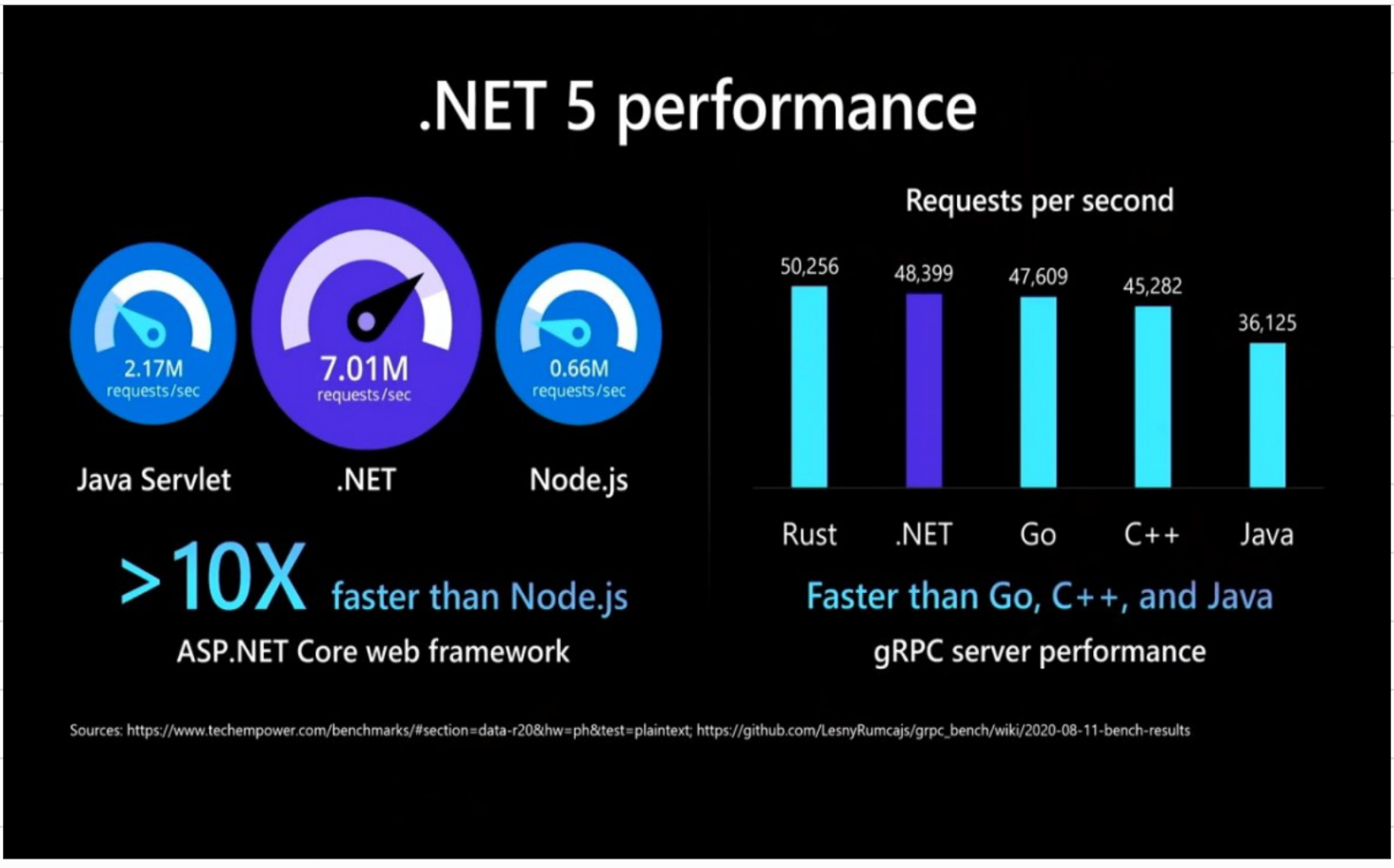


• NET – Runtime (like Node.js)

Express – ASP •NET core (like)

ADO.net – database connectivity for views

ASP.net mvc (model views controllers)

Language – C# (C-Sharp)

Power of C++
Simplicity of java
Elegance of Visual Basic

} •NET (open source from 2016) Launched in 2000

Latest till now – •NET 7.0 (12 Dec 2022)

☐ Download •NET

- .NET Core - compatible to Linux, Mac, Windows
- .NET Framework — windows based projects

if Visual Studio downloaded
↳ .NET comes automatically.

But if want to run in environment without visualstudio.
download .NET 7.0 SDK

To check .NET version
`dotnet --version`

.NET uninstallation tool

Sidebyside execution if 2versions of DOTNET are there in machine.

Uninstall the tool

Windows | macOS

1. Open **Add or Remove Programs**.
2. Search for `Microsoft .NET SDK Uninstall Tool`.
3. Select **Uninstall**.

Download the .NET SDK uninstall tool.

`list` command

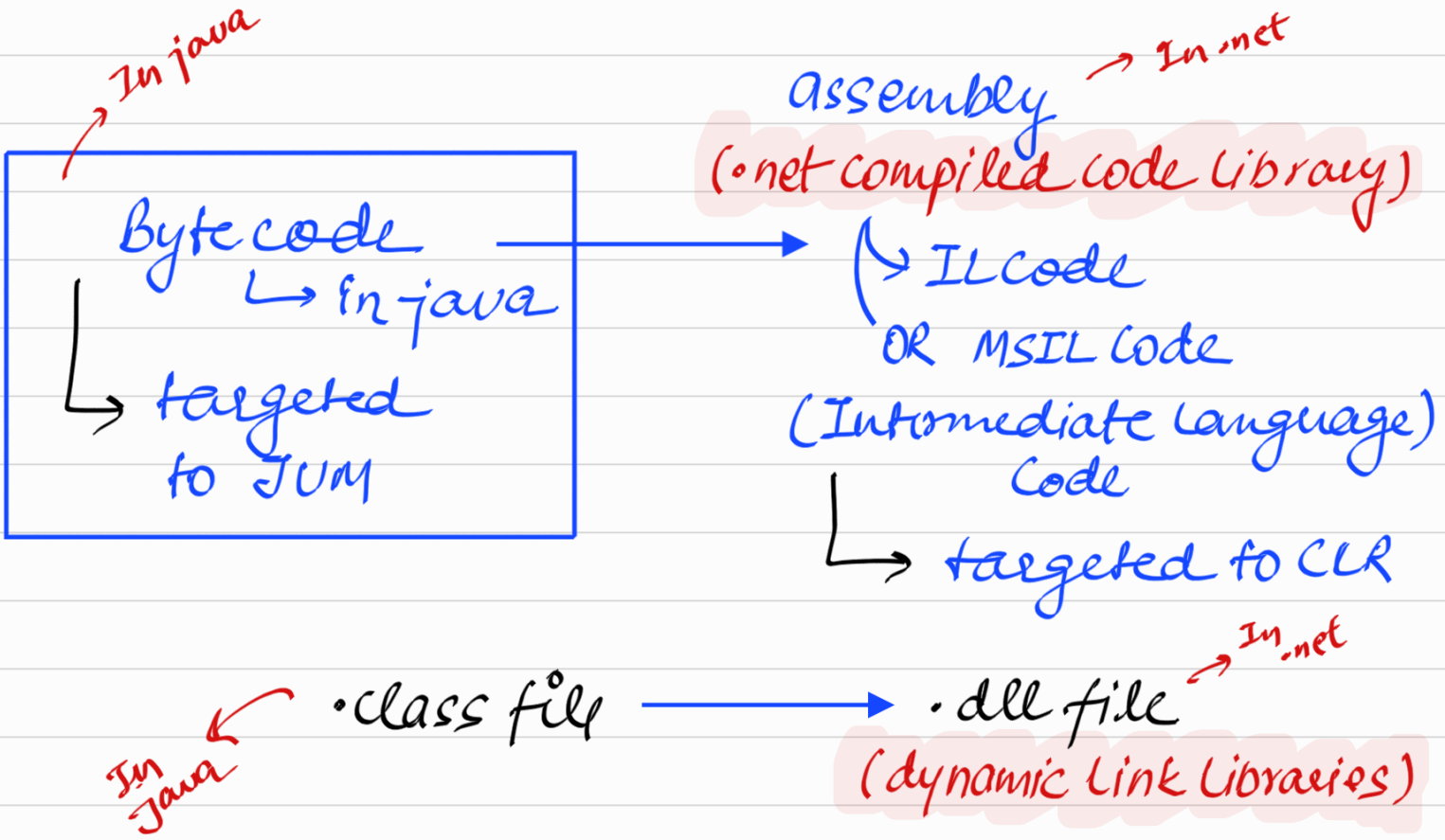
Synopsis

Console Copy

```
dotnet-core-uninstall list [options]
```

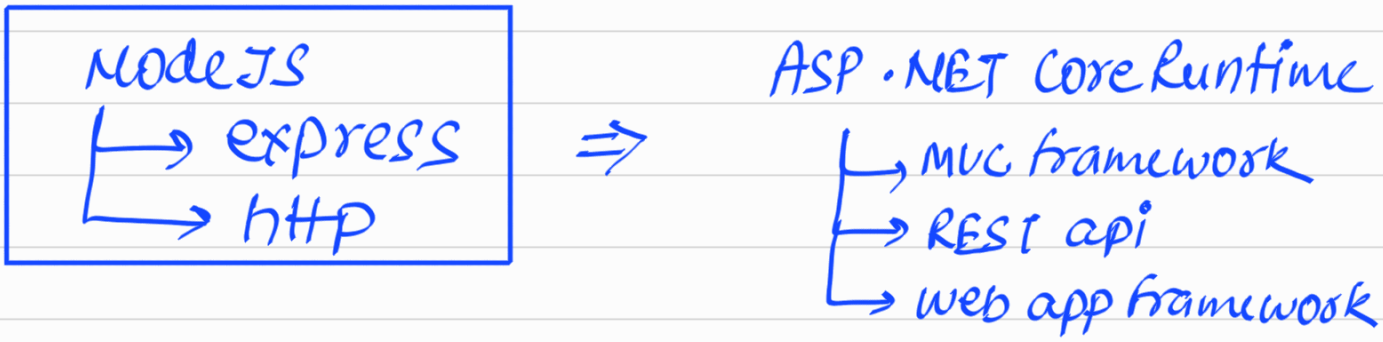
JVM → CLR
Java Virtual Machine = Common Language Runtime
(Execution engine for)

(Execution Engine for) .NET



In Java .class file → In .net .dll file (dynamic link libraries)

SDK contains
- .dll
- .exe (tools, readymade app.)



- .NET Windows desktop Runtime
  - --- Libraries for Windows GUI application.
  - --- 2D, 3D, multimedia

- .NET (command to make first application)

CLI Commands
Command line Interface

→ dotnet new console -o HelloworldApp

write
Console.WriteLine("Helloworld");
↘ goes nextline

program.cs
↳ By defeult acts as main.

string = String in Dotnet

string company = "Osho";

Console.write (company);

boolean = bool (same)

boolean status = false;
bool status = false;

for object creation

anonymous type

var p = new {
firstName = "Satish",

```
                                    LastName = "Dhavan"
                                };
Console.WriteLine(p.FirstName + "    " + p.LastName);
```

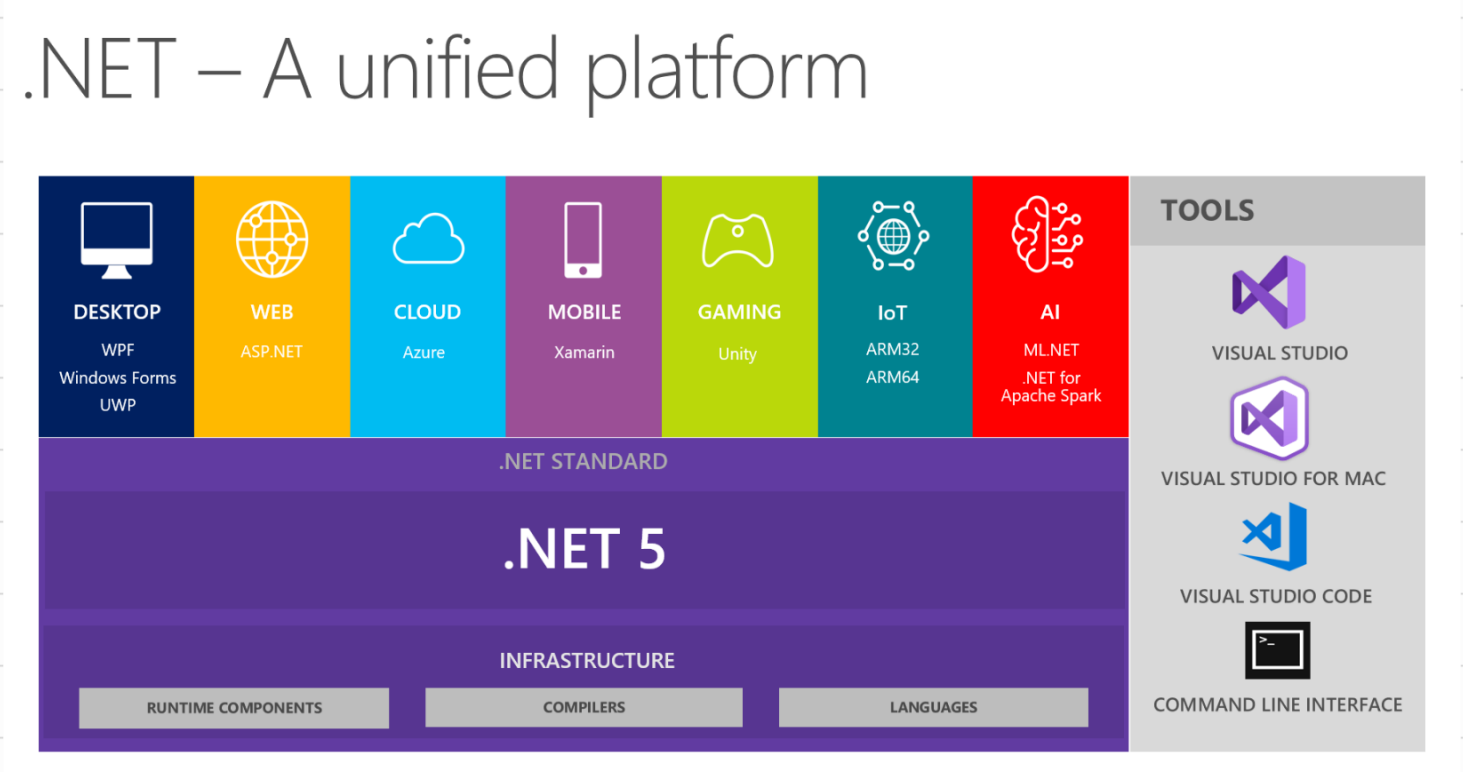
For Running Code:-

cmd → dotnet build
↳ in same folder.

↓

After build **bin** folder is created.

dotnet run
↳ to Run the app in Console.



in java ← package = namespace → in C#

declaring variables in java:-

- value types
  - primitive data types

- int, float, double, bool, struct, char
- enum

- Reference types
  - class
  - interface
  - delegate
  - event

} Not present in java

- collection framework
  - namespaces
    - ↓ System.Collections.Generic

CTS – Common Type System
↳ decides which are value type
or Reference type.

→ var student = "OSHO";
→ dynamic address = "Pune";
→ let arr;

Lambda function is also used in .net.
↳ Arrow function

C# is pure Object Oriented Programming Language.
Java is Not.
#multiple inheritance.

Wrapper class :-

Int64 , Int32

Ex. Int64 number = 36;

Account.cs

```
namespace Banking;

public class Account {

    private float balance = 5000;

    public void SetBalance(float amount){
        this.balance = amount;
    }

    public void GetBalance(){
        return this.balance;
    }

    public Account(){
        this.balance = 10000;
    }

    public Account(float amount){
        this.balance = amount;
    }
```

package like

setter

not necessary

getter

default constructor

Parameterised constructor

} constructor chaining

// In .net destructors are also there.
// destructor is always called by
// garbage collector before object
// is going to be removed from heap.

Chaining is also possible like java.

Deinitialization

If file is opened in Constructor then should be closed in destructor

```
~Account() {
    //
}
```

Exception :-

```
public void Withdraw(float amount) {
    if(amount == 0) {
        throw new Exception("Amount not zero");
    }
    this.balance -= amount;
}
```

in java ← Comiple = build → in .net

Using and Creating Object in Main :-

→ Import as java

```
using Banking;
```

Reference value ↑

→ Object creation

```
Account acc123 = new Account (60000);
acc123.Deposit (15000);

float currentBalance = acc123.GetBalance();

Console.WriteLine ("Curr Balance = {0}", currentBalance);
```

↳ placeholder at 0th pos

Using Generic :-

```
Using System.Collection.Generic;

List<account> accounts = new List<Account>();
accounts.add (acc123);
accounts.add (acc124);

// for each                    variable
                                  ↓
foreach (Account theAccount in accounts){
    float result = theAccount.GetBalance();
    Console.Writeline ("Current
                Balance = {0}", result);
}
```