Uttaranchal School of Computing Sciences

Application Development Using .NET Technology Framework

MCA -C551

Assignment I

SUBMITTED TO: -
Mr. Yogesh Juyal
(Assistant Professor)
USCS

SUBMITTED BY: -
Brijesh Sharma
Section: - MCA-A II
Roll No- 35

Q.1 :- What is .Net framework explain its all components with .Net framework diagram ?

• Net Framework :-

The .Net framework is a software development plateform developed by Microsoft for building & running application on windows. It provides a run-time environment, a set of class libraries, and tools to build web, desktop and mobile applications.

Component of .Net framework

① Common Language Runtime (CLR) :-

The runtime that executs .Net applications and provides services like memory management, garbage calleation & security.

② Base class, Library (BCL) :-

A calleation of reusable classes for common functionality, such as I/O, string manipulation and data access.

(iii) ADO.Net :-

Provides access to databases and data sources, enabling data manipulation and retrived.

(iv) ASP.Net :-

A framework for building dynamic web applications & services, including web forms, MVC, and web API.

(v) Window ~~Form (WinForms)~~ :- Presentation foundation (WPF)

A framework for building rich, graphics desktop application with advanced UI feature like 2D/3D graphics & animations.

(vi) Window Form (WinForm) :-

A framework for building desktop applications with graphical user interfaces (GUI).

(vii) Window Communication :-

A framework for building service-oriented application and enabling communication over network.

(viii) Entity Framework :-

An Object - Relational mapping (ORM) framework for working with database using .Net objects.

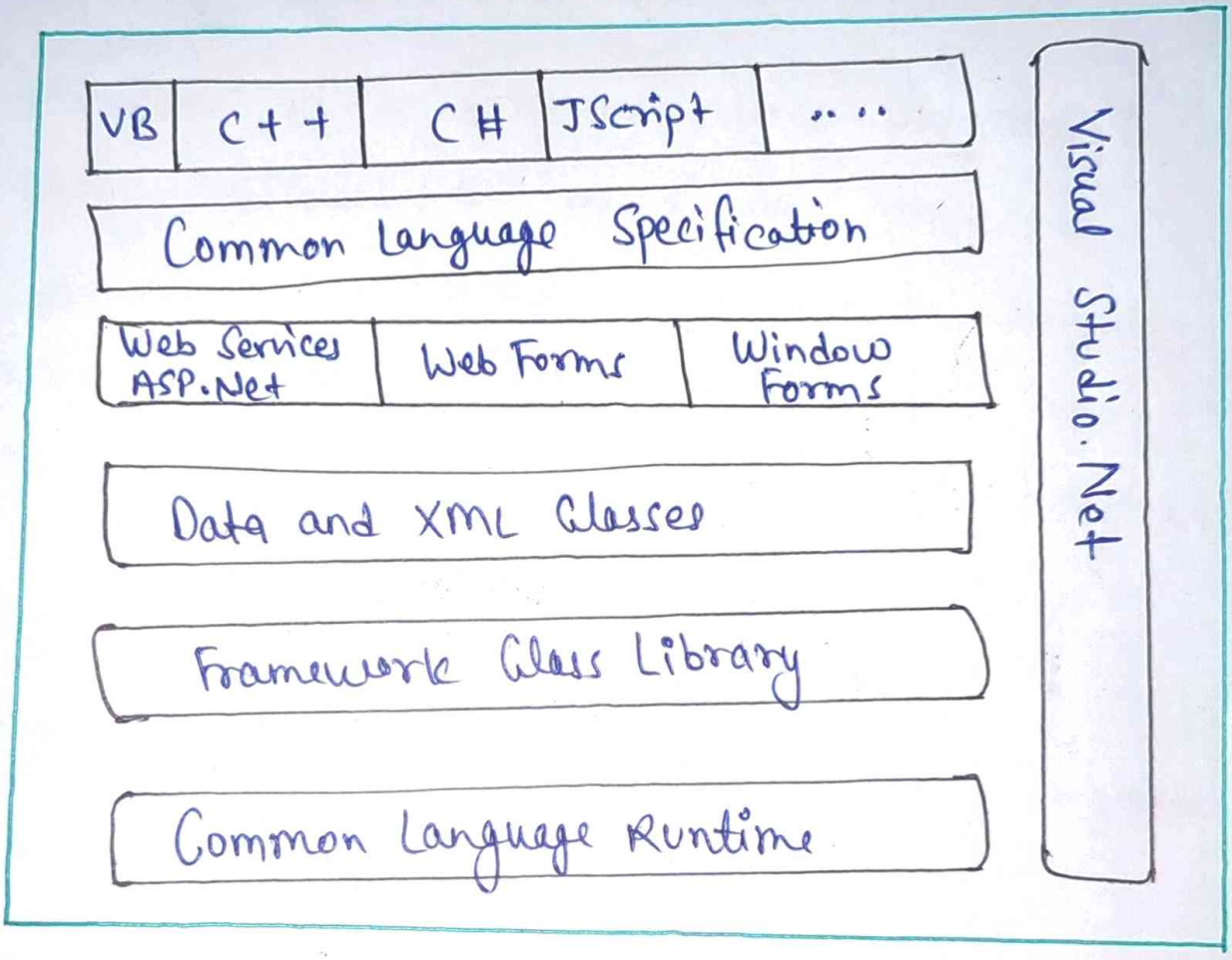
(ix) Common Type System (CTS) :-

Ensure that types in differenceent programming language can interact with each other.

(x) Global Assembly Cache (GAC) :-

A storage location for shared assemblies used by multiple applications.

.Net framework Diagram :-



(5)

Q.2:- Explain source program execution architecture in .Net Technology with its diagram.

- The execution of a ~~program~~ source program in .Net Technology follows a multi-step process. it involves compilation, Intermediate language conversion, just-in-time (JIT) compilation, and execution with the common language Runtime (CLR)

① Writing the Source Code.

- Developer write code in .Net - supported language like C#, VB.Net an f#.

② Compilation into Intermediate Language:-

- The .Net compiler compile the source code into microsoft intermediate language (MSIL) (also called Common Intermediate Language).
- This IL code is CPU - independent.

③ Metadata & Assembly formation:-

The compiled IL code is stored in an assembly along with metadata (which contained type defenetions, version information and security details.

- Assemblies can be private or shared.

④ Loading by Common Language Runtime (CLR):-

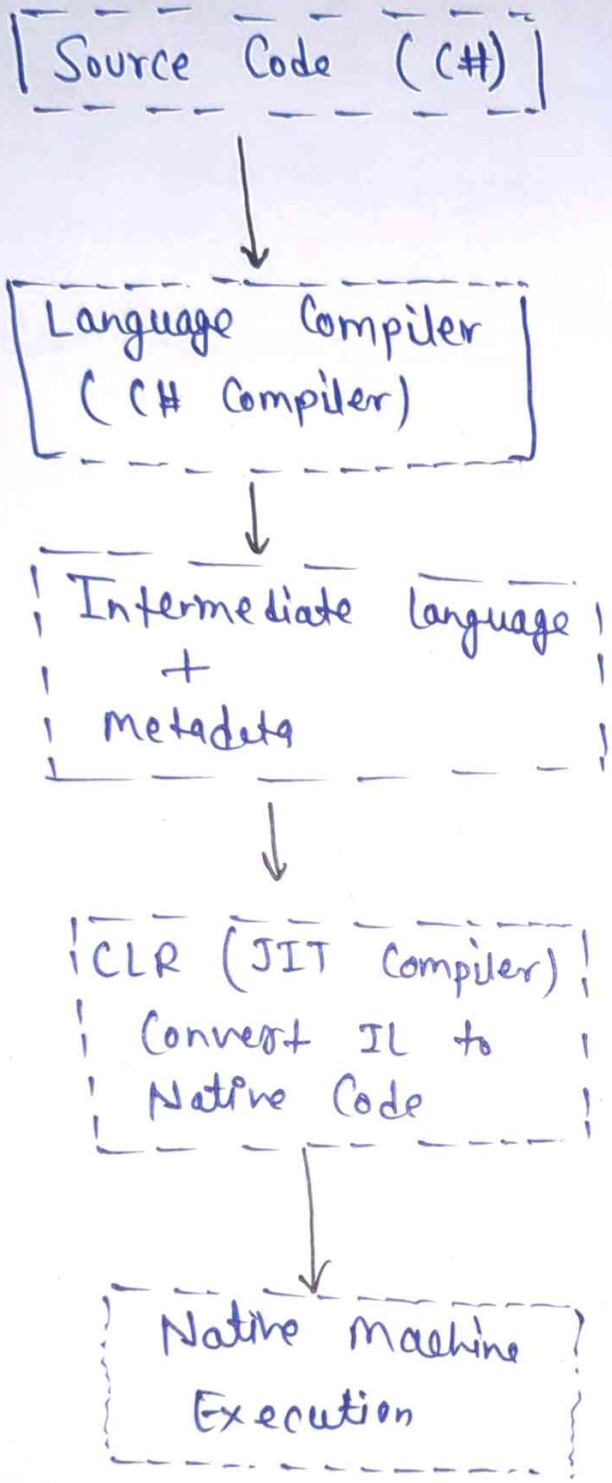
- The CLR loads the assembly into memory for execution.

⑤ Execution by CLR →

- The native code is then execution by the systems CPU within the managed. environment of the CLR
- The CLR provide additional services like:
- Memory management
- Garbage Collection
- Exception Handling
- Security Execution

Execution Flow

→ The developer writes code in a.Net language (C#, VB.Net, etc)
→ The compiler translate the code into CIL and create assembly.
→ When the program is executed, the CLR loads the assembly
→ The JIT Compiler within the CLR translates the CIL into native code for the specific plateform.
→ The CLR manages the execution of the native code.
→ The program interact with the FCL to access pre-built functionalities


Source Code (C#)
Language Compiler
(C# Compiler)
Intermediate language
+
Metadata
CLR (JIT Compiler)
Convert IL to
Native Code
Native Machine
Execution

Q.3:- Write a program in C# to find weather the entered number is prime number or not?

```
- using System;
  using System.Collection.Generic;
  using System.Text;

  public class PrimeNumberCheeker
  {
     public static void main (string[] args)
     {
        Console.WriteLine("Enter a number to check:");
        if (int.TryParse (Console.ReadLine(), out int number))
        {
           if (number <= 1)
           {
              Console.WriteLine("A number less than or
                 equal to 1 is not prime.");
           }
           else
              {
                 bool isPrime = isPrime (number);
                 if (isPrime)
                 {
                    Console.Writeline ($"{number} is not a prime
                                        number.");
           }
        }
     }
  }
```

```
else
 {
    Console.WriteLine ("Invalid input. Please enter a valid int.");
 }

    Console.ReadKey ();
}

 static bool IsPrime (int number)
{
    if (number <= 3)
    {
      return true;
    }
    if (number % 2 == 0 || number % 3 == 0)
    {
      return false;
    }
    for (int i= 5; i* i <= number; i= i +6)
    {
      if (number % i == 0 || number % (i+2)==0)
      {
        return false;
      }
    }
   return true;
}
```

Q.4 :- Write short notes on:

(a) CLR :- It is an important part of a .Net framework that work like a virtual component of the .Net framework to executes the different languages program like C#, Visual Basic, etc. A CLR also helps to convert a source code into the byte code, and this byte code is known as CIL and MSIL.

(b) Framework Base Class Library :-

The framework class library is a collection of reusable classes interfaces and types provide by the .Net framework.

- It includes libraries for file handling, database access (ADO.Net), networking collection, threading and UI components.

(c) CLS :- (Common Language Specification)

it is a subset of a common type system (CTS) that defines a set of rules and regulations which should be followed by every language that comes under the .net framework. In the other words, a CLS language should be cross-language interaction or ~~interper~~ interoperability.

CTS :- (Common Type System)

- It specific a standard that represent what type of data and value can be defined and managed in computer memory at runtime. A CTS insure that programming data defined in various language should be interact with each other to share information.

JIT (Just-In-Time)

JIT compilation in .Net is the process where the Common Language Runtime (CLR) converts intermediate language (IL) code into native machine code at the moment its needed during program execution.

MSIL (Microsoft Intermediat Language)

MSIL, also known as CIL is the intermediate code produced by .Net compilers from source code (like C# or VB.Net). it is a plateform-independent set of instructions that the Common Language Runtime convert into native machine code at runtime.

MSIL enables .Net's cross plateform capabilities.

Garbage Collection :-

Garbage Collection in .Net is the automatic memory management process where the common Language Runtime (CLR) reclaims memory occupied by object that are no longer being used by the application.

Q.5 :- Explain all the features of C# programming with it benefits.

- C# is a modern, object oriented programming language developed by Microsoft as part of .Net framework.

- Key feature of C#

① Object-Oriented Programming (OOps) - Support OOps principal

- Encapsulation - Building data and methods in single unit.
- Inheritance : Enbaling code reuse by allowing classes to inherit from other class.
- Polymorphism :- Allowing methods to take multiple forms.
- Abstraction :- Hiding implementation details from user.

② Type Safety :-
C# is a strongly typed language, meaning variables must be declared with a specific data type, reduce runtime error.

③ Automatic memory Management (Garbage Collection) :-
C# includes an automatic garbage collection that manage memory allocation and declaration, reducing the risk of memory leaks.

④ Rich Library Support :- The .Net framework provides a vast collection of libraries (file handling, database connectivity, networking, etc)

⑤ Interoperability!- allow interaction with other language like C++ and JavaScript.

⑥ Exception Handling!- C# has a structured exception handling mechnism using try, catch, finally and throw statements, making application more robust.

⑦ Plateform Independent !-

With .Net Core and .Net 5+, C# can run on multiple platform, including Windows, Linux, and MacOS.

⑧ Security Features !- features like type safety, code access security, and role-based ~~authati~~ ~~auth~~ authntication to prevent unauthorized access.

⑨ Multi-Threading Support!-

C# supports multi-threading, allowing application to run multiple tasks simultaneously improving efficiency.

⑩ Dynamic Features!-

C# includes the Dynamic ~~feature~~ keywork, enabling runtime type checking and interaction with dynamic languages like python.

(11) Delegates and Events :- C# support ~~defining properties and indexers~~ event-driven programming using delegates and events, making it useful for GUI applications and event driven system. (14)

(12) Cross-Plateform Development :- With .Net framework Core and .Net 5+, developers can build application that run on different operating system.

Benefits of C#

- Easy to learn & use.
- High Performance
- Versatility
- Strong Community Support
- Integration with .Net framework
- Backed by Microsoft
- Secure and Reliable
- Scalability and maintainbility
- Extensive Tooling Supports
- Supports for modern Programing paradigms.