AKDENIZ UNIVERSITY
FACULTY OF ENGINEERING
COMPUTER SCIENCE
&
ENGINEERING DEPARTMENT

PROGRAMMING LANGUAGES
“Programming Language Evaluation Report”
“C# Programming Language”

Mustafa Onur Başer 20170808008
Yusuf Çelik 20170808006

INSTRUCTOR: ASST.PROF.DR. MURAT AK

History

C# is one of the most popular programming languages in the world. C# is a general-purpose, modern, type-safe and object-oriented programming language that runs on .NET Framework. C# is an object-oriented programming language but also it includes support for component-oriented programming as well. In January 1999 a team formed in Microsoft for developing a new language called "Cool" which stood for "C-like Object Oriented Language". In July 2000 .NET announced at the Profssional Developers Conference and the language had been renamed as C#. Since the team is formed by Anders Hejlsberg he is considered to be the author of the C# but it was a team work. He is currently the lead architect of C# in the Microsoft. He is also the author of the Turbo Pascal and the chief architect of the Delphi. C# had very influenced by other languages but the main ones are C, C++ and Java so that means C# will be familiar to the developers that Works on those languages but C# was created from the ground up. These influences also came with some speculations as well but they were mostly about the similarities with the Java. The creator of the Java James Gosling and Bill Joy the co-founder of Sun Microsystems, the originator of the Java, called C# an "imitation" of Java. And Gosling also said that "C# is sort of Java with reliability, productivity and security deleted.". The authors of a C++ streams book Klaus Kreft and Angelika Langer stated in a blog that " Java and C# are almost identical programming languages. Boring repetition that lacks innovation." , " Hardly anybody will claim that Java or C# are revolutionary programming languages that changed the way we write programs." and " C# borrowed a lot from Java and vice versa. Now that C# supports boxing and unboxing, we'll have a very similar feature in Java.". But in July 2000 Hejlsberg said that C# is "not a Java clone" and is "much closer to C++" in its design. These similarities and influences also seen in the languages name; "++" indicates that a variable should be incremented by one after being evaluated in C language because of that reason "++" in the C++ means to more than C and the "#" on the C# implyies that the language is an increment of C++ since the "#" represents four "+" symbols in a two-by-two grid. But after these speculations with the release of the C# 2.0 in November 2005, C# and Java evolved on increasingly divergent trajectories, becoming a two quite different languages. But still they have some similarities. And C# is also approved by the European Computer Manufacturers Association (ECMA) and International Standarts Organization (ISO) as well. Now it is considered one of the most popular programming languages. In 2019 it is noted that 6.7 users were using C#.

Versions

Version	.NET Framework	Visual Studio	Year
C# 1.0	.NET Framework 1.0/1.1	Visual Studio .NET 2002	January 2002
C# 2.0	.NET Framework 2.0	Visual Studio 2005	November 2005
C# 3.0	.NET Framework 3.0\3.5	Visual Studio 2008	November 2007
C# 4.0	.NET Framework 4.0	Visual Studio 2010	April 2010
C# 5.0	.NET Framework 4.5	Visual Studio 2012/2013	August 2012
C# 6.0	.NET Framework 4.6	Visual Studio 2013/2015	July 2015
C# 7.0	.NET Core 2.0	Visual Studio 2017	March 2017
C# 7.1	.NET Core 2.0	Visual Studio 2017 version 15.3	August 2017
C# 7.2		Visual Studio 2017 version 15.5	November 2017
C# 7.3	.NET Core 2.1 .NET Core 2.2 .NET Framework 4.8	Visual Studio 2017 version 15.7	May 2018
C# 8.0	.NET Core 3.0	Visual Studio 2019	September 2019

New features with the all new versions

C# 1.0

- Classes
- Structs
- Interfaces
- Events
- Properties
- Delegates
- Expressions
- Attributes
- Literals
- Basic features

C# 2.0

- Generics
- Partial types
- Anonymous methods
- Iterators
- Nullable value types
- Getter/setter separate accessibility
- Method group conversions (delegates)
- Co- and Contra-variance for delegates
- Static classes
- Delegate inference
- Null coalescing operator

C# 3.0

- Implicitly typed local variables
- Object and collection initializers
- Auto-Implemented properties
- Anonymous types
- Extension methods
- Query expressions
- Lambda expressions
- Expression trees
- Partial methods

C# 4.0

- Dynamic binding
- Named and optional arguments
- Generic co- and contravariance
- Embedded interop types ("NoPIA")

C# 5.0

- Asynchronous methods
- Caller info attributes

C# 6.0

- Compiler-as-a-service (Roslyn)
- Import of static type members into namespace
- Exception filters
- Await in catch/finally blocks
- Auto property initializers
- Default values for getter-only properties
- Expression-bodied members
- Null propagator (null-conditional operator, succinct null checking)
- String interpolation
- nameof operator
- Dictionary initializer

C# 7.0

- Inline out variable declaration
- Pattern matching
- Tuple types and tuple literals
- Deconstruction
- Local functions
- Digit separators
- Binary literals
- Ref returns and locals
- Generalized async return types
- Expression bodied constructors and finalizers
- Expression bodied getters and setters
- Throw can also be used as expression

C# 7.1

- Async main
- Default literal expressions
- Inferred tuple element names

C# 7.2

- Reference semantics with value types
- Non-trailing named arguments
- Leading underscores in numeric literals
- private protected access modifier

C# 7.3

- Accessing fixed fields without pinning
- Reassigning ref local variables
- Using initializers on stackalloc arrays
- Using fixed statements with any type that supports a pattern
- Using additional generic constraints

C# 8.0

- readonly struct members
- Default interface members
- switch expressions
- Property, Tuple, and positional patterns
- using declarations
- static local functions
- Disposable ref struct
- Nullable reference types
- Indices and Ranges
- Null-coalescing assignment
- Async Streams

Why Microsoft invented C# ?

We talked about the history of C# but why did Microsoft decided to develope a programming language, lets take a look into Microsofts reason to develop the C#. The reason is not a technological concern it is like a Microsofts company policy. When we look in Microsoft history they weren't the lead developer for popular services like Excel situation, Lotus Software developed Lotus 1-2-3 which is a spreadsheet program. Then Microsoft decided to make Excel. Netscape and Mosaic battled for the first web browser and in 1993 NCSA Mosaic is developed. After that in 1995 Microsoft came with Internet Explorer. In 1995 also Java is introduced by the Sun Microsystems, Microsoft saw the potential in the Java so Microsoft came out with some specific extension to Java to make it "easier for developers" and make it "run faster on Windows". But this blowed back to Microsoft and in 1997 Sun sued Microsoft. Because if they keep coming with these extension it couldn't be called Java so this forced Microsoft to discontinue its implementations. But Microsoft decidede to "out Sun" Sun, by introducing their own programming language and platform, and effectively killing Java on Windows. So we come to the renowned programming language designer Anders Hejlsberg and his developer team. Hejlsberg already had experience mutating and improving existing programming languages so Microsoft gave him the task of the creating a "better Java". Microsof and the Hejlsbergs team had some objectives about C#: to be simple, modern and object-oriented. Main focus of the team was taking things about existing languages and adding improvements to make something better. Thus C# were born along with the .NET framework.

Why to Use and Learn C# ?

There are few reasons that you should use and learn it. For learning part, C# is considered as beginner friendly and easy to start with. Because it is developed to be simple and easy to use, you will get a ton of materials on the internet that can help you with your problems in your learning and developing process as well. This amount of resources includes boks, tutorials, videos and much more. When we look at the developing part the best reason to use C# is you can have the ability to develop many kinds of applications for many different plaftorms such as web, desktop, mobile, games an deven robotics. So that means you can use it in any different platforms which mean you won't have to learn different languages for different platforms. It can be used for building Windows client applications, web applications, native iOS and Android mobile apps, games, Cloud and Azure apps, Blockchain apps and much more. C# is a statically-typed language so your code will be checked before it gets built in an app. Error tracking will be easier because of its type. And codebase will be consistent and eaisier to maintain as it grows in size and complexity. C# is also faster then dynamically-typed languages because things are more clearly defined. Like other object-oriented languages, C# also includes features like encapsulation, polymorphism and inheritance. C# also has powerful development tools as well, which we will mention later how to set up them for different platftorm. Microsoft has many powerful tools for programmers to develop better programs. The most popular and important one is Visual Studio and .NET Core 3.0 is one of the fastest platforms among all all the other software development platforms. C# also an open-source programming language as well, its compiler is available on Github. And most importantly Microsoft got your back. They are still adding new features to the C# and provided support for seamless integration with other Microsoft Technologies. They also provide a training website called Microsoft's Virtual Academy that contains many courses. There are also other popular websites like Pluralsight, LearnCS.org and Complete C# Tutorial.

Specifig things in C#

Back in the days, around .NET 3 is released, Microsoft came up with really cool advantages:

LINQ (Language Integrated Query), Entity Framework, Properties, ASP.NET specialties and robust Getters/Setters for classes.

Nowdays, Languages like Java and Python have all these features as well. Some of them are external libraries, but they're still there.
Other languages cath up with the C#.

But C# is still amazing.

Fun, robust syntax.
Development is fast, with the help of right IDE.
Can be used for many different fields like: desktop, web, mobile, game, cloud and IoT.
ASP.Net is the fastest web framework out there.
Visual Studio and Visual Studio Code are great and powerful IDEs.
There are a lot of modern technologies built on top of C#, like Xamarin.
One of the biggest tech company Microsoft is backing up the C# since they developed it

C# basics

```
using System;

namespace CSHomework
{
   class Program
   {
      static void Main(string[] args)
      {
         string a = "hello";            //We create string in here
         int ab= 4984;                  // Create integer value
         double x = 5.75;              //Double value

         Console.WriteLine(a);  //Write hello

         int add = 6555 + 7564;

         int subtraction = 100 - 21;

         int multiple = 50 * 26;
```

```
            int division = 50 / 5;

            int mod = 406 % 20;

            string[] car; // declare array but there is not length of array
            car = new string[4]; // now we have length of array

            car[0] = "Toyota";      //index zero is toyota
            car[1] = "Honda";     //index one is honda
            car[2] = "Opel";      //index two is opel
            car[3] = "Ferrai";   //index three is ferrari

            int[] num = { 1, 5, 6, 8, 41, 81, 94, 28, 4 };


//--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

            if (division == 10) // inside of if must be a boolean we can't enter other varible type like integer
double if divison is equals to 10 writes "cx"
            {
                Console.WriteLine("cx");
            }
            else if (division == 4) // if divison is not equals 10 but equals 4 writes "50 / 5 = 4"
            {
                Console.WriteLine("50 / 5 = 4");
            }

            else Console.WriteLine("No"); //if division is not 10 or 4 writes "No"

//--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

            for(int i = 0; i < 10 ; i++) { Console.WriteLine(i); } //Writes numbers 0 to 9

            int xx = 0;
            while (xx<10) { Console.WriteLine(x); xx++;}// Same as upside for loop


        }
    }
    }
```

C# Methods

```
static void Main(string[] args)
        {
            print3();//This is method call
        }

        public static int add(int a, int b)//This is method retuns integer value and this method has
parameters
        {
            return a + b;
        }

        public static int add(int a, int b, int c) //This is overloading in this class we have another add method
but in this method we have 3 parameter a b c  other add method 2 parameter
        {
            return a + b + c;
        }
        public static void print3() { Console.WriteLine("Result is" + add(5, 2, 3)); } //This method returns
nothing because this method has a void keyword and also this method has not got parameter.
```

C# OOP

Classes

```
 class Car
    {
        private int speed; //private fields we can't access to the speed from other classes because private we
must use getter setter setter is optional we can also use constructor for setter
        private String name;
        private int year;
        private String color;
        private int doorNumber;
        private int price;

        public Car(int speed, String name, int year, String color) //Constructor with parameter
        {
            this.speed = speed;
            this.name = name;
            this.year = year;
            this.color = color;

        }

        public int Speed //Getter
        {
            get { return speed; }
        }
        public String Name //Getter
        {
            get { return name; }
        }
        public int Year //Getter
        {
            get { return year; } //Getter
        }
        public String Color //Getter
        {
            get { return color; }
        }

        public int DoorNumber //Getter and setter
        {
            get {return doorNumber; }
            set { doorNumber = value; }
        }

        public int Price //Automatic getter and setter
        { get;   set; }

    }
}
class Program
    {
        static void Main(string[] args)
        {
            Car toyota = new Car(300, "Toyota AE86", 1972, "White");//Send paramaters to the constructor
            toyota.DoorNumber = 4;  //Setter
            toyota.Price = 20000;  //Setter
Console.WriteLine("Car name is " + toyota.Name + " speed: " + toyota.Speed + " year: " + toyota.Year + "
color: " + toyota.Color + " door number: " + toyota.DoorNumber + " price: " + toyota.Price);//Getters
 }}
```

Inheritance

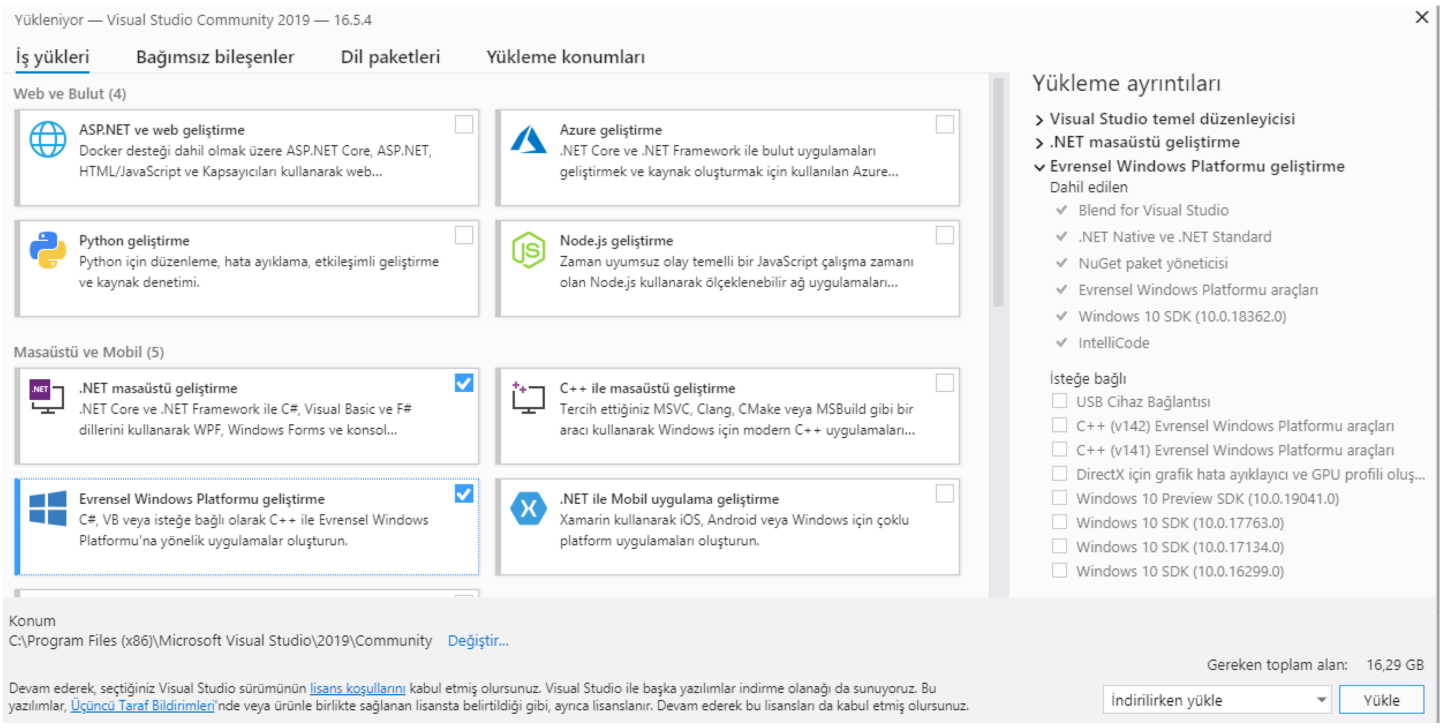
```
   class Animal
    {
        private String voice;
        private int age;
        private String animalType;

         public int Age
        { get; set; }
        public String Voice
        { get; set; }
        public String AnimalType
        { get; set; }

        public void sound(String voice , String animalType)
        {
            this.animalType = animalType;
            this.voice = voice;
            Console.WriteLine(this.animalType + " says " + voice);
        }
    }
}
class Lion : Animal
    {
 public Lion()
     {
            AnimalType = "Lion";
            Age = 3;
            Voice = "Roar";
        }
    }
   class Program
    {
        static void Main(string[] args)
        {
            Lion lion = new Lion();
```

lion.sound(lion.Voice, lion.AnimalType) }}}

Enviroments for C# and How to Setup Them

Best and the most popular IDE for C# is Visual Studio. It is available for both Windows and macOs. It can be downloaded from Microsofts website. .NET is one of the most popular enviroments for C# and it started its journey with C# ass well. .NET is a free, cross-platform, open-source developer platform for building web, mobile, desktop, microservices, game , machine learning, cloud and IoT apps. It is available for Windows, Linux and macOS on the Microsofts website. Native iOs and Android mobile apps can be build be Xamarin and it can be added to Visual Studio. Visual Studio Code is another tool by Microsoft. It is a code editor for multiple languages including C#. It is also available for Windows, Linux and macOS on the Microsofts website. There is also a cross-platform, open-source framework for building web apss and services with .NET and C# called ASP.NET. It is available for Windows, Linux and macOS on the Microsofts website. For the game development part there is Unity. It is a game development platform for building 2D and 3D games with .NET and C#. It is available for Windows, Linux and macOS on the Unity's website.

Windows:
1)https://visualstudio.microsoft.com/tr/downloads/
2)Download Visual Studio.
3)Run exe file.
4)


Yükleniyor — Visual Studio Community 2019 — 16.5.4

İş yükleri | Bağımsız bileşenler | Dil paketleri | Yükleme konumları

Web ve Bulut (4)

ASP.NET ve web geliştirme
Docker desteği dahil olmak üzere ASP.NET Core, ASP.NET, HTML/JavaScript ve Kapsayıcıları kullanarak web...

Azure geliştirme
.NET Core ve .NET Framework ile bulut uygulamaları geliştirmek ve kaynak oluşturmak için kullanılan Azure...

Python geliştirme
Python için düzenleme, hata ayıklama, etkileşimli geliştirme ve kaynak denetimi.

Node.js geliştirme
Zaman uyumsuz olay temelli bir JavaScript çalışma zamanı olan Node.js kullanarak ölçeklenebilir ağ uygulamaları...

Masaüstü ve Mobil (5)

✓ .NET masaüstü geliştirme
.NET Core ve .NET Framework ile C#, Visual Basic ve F# dillerini kullanarak WPF, Windows Forms ve konsol...

C++ ile masaüstü geliştirme
Tercih ettiğiniz MSVC, Clang, CMake veya MSBuild gibi bir aracı kullanarak Windows için modern C++ uygulamaları...

✓ Evrensel Windows Platformu geliştirme
C#, VB veya isteğe bağlı olarak C++ ile Evrensel Windows Platformu'na yönelik uygulamalar oluşturun.

.NET ile Mobil uygulama geliştirme
Xamarin kullanarak iOS, Android veya Windows için çoklu platform uygulamaları oluşturun.

Yükleme ayrıntıları
> Visual Studio temel düzenleyicisi
> .NET masaüstü geliştirme
∨ Evrensel Windows Platformu geliştirme
Dahil edilen
✓ Blend for Visual Studio
✓ .NET Native ve .NET Standard
✓ NuGet paket yöneticisi
✓ Evrensel Windows Platformu araçları
✓ Windows 10 SDK (10.0.18362.0)
✓ IntelliCode
İsteğe bağlı
☐ USB Cihaz Bağlantısı
☐ C++ (v142) Evrensel Windows Platformu araçları
☐ C++ (v141) Evrensel Windows Platformu araçları
☐ DirectX için grafik hata ayıklayıcı ve GPU profili oluş...
☐ Windows 10 Preview SDK (10.0.19041.0)
☐ Windows 10 SDK (10.0.17763.0)
☐ Windows 10 SDK (10.0.17134.0)
☐ Windows 10 SDK (10.0.16299.0)

Konum
C:\Program Files (x86)\Microsoft Visual Studio\2019\Community Değiştir...

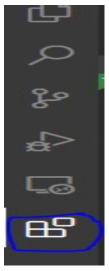
Gereken toplam alan: 16,29 GB

Devam ederek, seçtiğiniz Visual Studio sürümünün lisans koşullarını kabul etmiş olursunuz. Visual Studio ile başka yazılımlar indirme olanağı da sunuyoruz. Bu yazılımlar, Üçüncü Taraf Bildirimleri'nde veya ürünle birlikte sağlanan lisansta belirtildiği gibi, ayrıca lisanslanır. Devam ederek bu lisansları da kabul etmiş olursunuz.

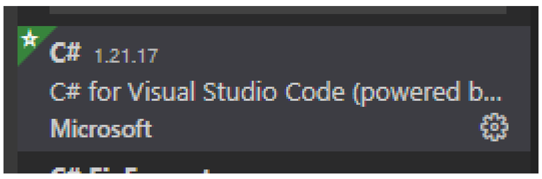
İndirilirken yükle | Yükle

Press Install(Yükle).
Then it will installing to our computer.

Ubuntu
There is not visual studio for ubuntu but we can download Visual Studio Code for Ubuntu
1)Open the terminal
2)Write sudo snap install code --classic
3)It will download Visual Studio Code
4)In vs code we must press blue square



5)Final step we must install C# for Visual Studio Code



C# 1.21.17
C# for Visual Studio Code (powered b...
Microsoft

Mac
https://www.youtube.com/watch?v=DS4zGjyo4Zs

Example Codes

1) Calculator
```
using System;

namespace Calculator
{
    class Calculator
    {
        static void Main(string[] args)
        {
            int exit = 0;
            print("Welcome To The Calculator");

            while (exit!=1)
            {
                int selection;
                print("For Addition press 1\nFor Substraction press 2\nFor Division press 3\nFor Multiplication
press 4\nFor Mode press 5");
                selection = Convert.ToInt32(Console.ReadLine());

                switch (selection)

                {
                    case 1:
                        print("Are you add 2 or 3 number?");
                            int twoOrThree = Convert.ToInt32(Console.ReadLine());
                        if (twoOrThree == 2)
                        {
```

```
            print("Please enter 2 number for addition");
            int x1 = Convert.ToInt32(Console.ReadLine());
            int y1 = Convert.ToInt32(Console.ReadLine());
            print("Your result is " + add(x1, y1));
        }
        else if (twoOrThree == 3)
        {
            print("Please enter 3 number for addition");
            int x1 = Convert.ToInt32(Console.ReadLine());
            int y1 = Convert.ToInt32(Console.ReadLine());
            int z1 = Convert.ToInt32(Console.ReadLine());
            print("Your result is " + add(x1, y1, z1));
        }
        break;
    case 2:
        print("Are you substract 2 or 3 number?");
        int twoOrThree2 = Convert.ToInt32(Console.ReadLine());
        if (twoOrThree2 == 2)
        {
            print("Please enter 2 number for substraction");
            int xx = Convert.ToInt32(Console.ReadLine());
            int yy = Convert.ToInt32(Console.ReadLine());
            print("Your result is " + subs(xx, yy));
        }
        else if (twoOrThree2 == 3)
        {
            print("Please enter 3 number for substraction");
            int xx = Convert.ToInt32(Console.ReadLine());
            int yy = Convert.ToInt32(Console.ReadLine());
            int zz = Convert.ToInt32(Console.ReadLine());
            print("Your result is " + subs(xx, yy, zz));
        }
        break;
    case 3:
        print("Are you divide 2 or 3 number?");
        int twoOrThree3 = Convert.ToInt32(Console.ReadLine());
        if (twoOrThree3 == 2)
        {
            print("Please enter 2 number for divison");
            int xxx = Convert.ToInt32(Console.ReadLine());
            int yyy = Convert.ToInt32(Console.ReadLine());
            print("Your result is " + division(xxx, yyy));
        }
        else if (twoOrThree3 == 3)
        {
            print("Please enter 3 number for division");
            int xxx = Convert.ToInt32(Console.ReadLine());
            int yyy = Convert.ToInt32(Console.ReadLine());
            int zzz = Convert.ToInt32(Console.ReadLine());
            print("Your result is " + division(xxx, yyy, zzz));
        }
        break;
    case 4:
        print("Are you multiply 2 or 3 number?");
        int twoOrThree4 = Convert.ToInt32(Console.ReadLine());
```

```
                if (twoOrThree4 == 2)
                {
                    print("Please enter 2 number for multiplication");
                    int xxxx = Convert.ToInt32(Console.ReadLine());
                    int yyyy = Convert.ToInt32(Console.ReadLine());
                    print("Your result is " + multiple(xxxx, yyyy));
                }
                else if (twoOrThree4 == 3)
                {
                    print("Please enter 3 number for multiplication");
                    int xxxx = Convert.ToInt32(Console.ReadLine());
                    int yyyy = Convert.ToInt32(Console.ReadLine());
                    int zzzz = Convert.ToInt32(Console.ReadLine());
                    print("Your result is " + multiple(xxxx, yyyy, zzzz));
                }
                break;
            case 5:
                print("Please enter 2 number for remaining ");
                    int x = Convert.ToInt32(Console.ReadLine());
                    int y = Convert.ToInt32(Console.ReadLine());
                    print("Your result is " + mod(x, y));


                break;
        }
        print("Do you want to exit if you want to exit please write 1");
        exit = Convert.ToInt32(Console.ReadLine());
        if (exit == 1) break;


    }

}

private static void print(String x)
{
    Console.WriteLine(x);
}

public static int add(int x, int y)
{
    return x + y;
}

public static int add(int x, int y, int z)
{
    return x + y + z;
}


public static int subs(int x, int y)
{
    return x - y;
}
```

```
        public static int subs(int x, int y, int z)
        {
            return x - y - z;
        }

        public static int multiple(int x, int y)
        {
            return x * y;
        }

        public static int multiple(int x, int y, int z)
        {
            return x * y * z;
        }

        public static int division(int x, int y)
        {
            return x / y;
        }
        public static int division(int x, int y, int z)
        {
            return x / y / z;
        }
        public static int mod(int x, int y)
        {
            return x % y;
        }

    }
}
```

2)Hospital

```
 class main
    {
        static void Main(string[] args)
        {
            Hospital hospital = new Hospital("A", "242 3226592");

            Nurse nurse = new Nurse("Ahmet", 1);
            nurse.addNurse(nurse);



            Doctor doctorAli = new Doctor("Ali","heart",48,1);
            doctorAli.addDoctor(doctorAli);

            Patient ali = new Patient("Ali", 1, 25, "Heart Attack");
            ali.addPatient(ali);
            ali.getPatient(1);

            Console.WriteLine(hospital.HospitalName);
        }
    }
}
```

```
class Hospital
    {
        private String hospitalName;
        private String telNum;

        public Hospital(String hospitalName, String telNum)
        {
            this.hospitalName = hospitalName;
            this.telNum = telNum;
        }

        public String HospitalName { get { return hospitalName; } }
        public String TelNum { get { return telNum; } }
    }
 class Patient
    {
        private String name;
        private int patientNumber;
        private int roomNo;
        private String disease;
        List<Patient> pati = new List<Patient>();

        public Patient()
        {
            Console.WriteLine("Please enter patient");
        }

        public Patient(String name, int patientNumber, int roomNo, String disease)
        {
            this.name = name;
            this.patientNumber = patientNumber;
            this.roomNo = roomNo;
            this.disease = disease;

        }

        public void addPatient(Patient patient)
        {
           pati.Add(patient);
        }

        public void getPatient(int patientNum)
        {
            int j = 0;
            for (int i = 0; i < pati.Count; i++)
            {
                if (pati[i].patientNumber == patientNum) j = i;

            }
            Console.WriteLine("Patient name:" + pati[j].name + " patient disase:" + pati[j].disease +" room
number:"+ pati[j].roomNo);

        }

        public void removePatient(Patient patient)
        {
            pati.Remove(patient);
        }

    }
  class Doctor
    {
        private String name;
        private String section;
        private int age;
        private int docNumber;
        List<Doctor> doc = new List<Doctor>();
        public Doctor()
        {
            Console.WriteLine("Please enter Doctor");
        }
        public Doctor(String name, String section, int age, int doctorNumber)
        {
            this.name = name;
            this.section = section;
            this.age = age;
            this.docNumber = doctorNumber;

        }
        public void addDoctor(Doctor doctor)
        {
            this.doc.Add(doctor);
        }

        public void getDoctor(int doctorNumber)
        {
            int j = -1;

            for (int i = 0; i < doc.Count; i++)
            {
                if (doc[i].docNumber == doctorNumber) j = i;

            }
            Console.WriteLine("Doctor's name:" + doc[j].name + " Doctor's number" + doc[j].docNumber +
"Doctor's age" + doc[j].age);


        }

        public void fireDoctor(Doctor doctor)
        {
            doc.Remove(doctor);
        }
    }
  class Nurse
    {
        private String name;
        private int no;
        List<Nurse> List = new List<Nurse>();
        public Nurse(String name, int no)
        {
            this.name = name;
            this.no = no;

        }

        public void addNurse(Nurse nurse)
        {
            List.Add(nurse);
        }

        public void getNurse(int nurseNumber)
        {
            int j = -1;

            for (int i = 0; i < List.Count; i++)
            {
                if (List[i].no == nurseNumber) j = i;

            }


            Console.WriteLine("Nurse name:" + List[j].name + "Nurse no" + List[j].no);


        }

        public void fireNurse(Nurse nurse)
        {
            List.Remove(nurse);
        }
    }
```

Java vs C#

Java vs C#

#1. Paradigm

Java	C#
Class-based, Object-Oriented language derived from C++	Object-Oriented, component-oriented, functional, strong typing.

#2. Application

Java	C#
Complex web based, highly concurrent application.	Web and game development, popular for mobile development.

#3. Project

Java	C#
Suited for complex web based concurrency project.	Best suited for game development projects.

#4. Usage

Java	C#
Messaging, web application, highly concurrent application.	Games, mobile development, virtual reality.

#5. Installation

Java

Require JDK (Java Development Kit) to run Java.

C#

.NET framework provides huge library of codes used by C#

#6. Scope

Java

Dominate server-side interaction.

C#

Server-side language with good programming foundation.

#7. Cross Platform

Java

Java is highly efficient for cross platform with its byte code.

C#

Compare to Java, C# needs to improve on this feature.

#8. Tools

Java

Eclipse, NetBeans, IntelliJ IDEA.

C#

Visual studio, MonoDevelop, #develop.