

بسم الله الرحمن الرحيم

# C#

Programming Language

MH. Movasaghinia

# Contact:

Mohammad Hosein Movasaghinia

Computer Engineering Student



[mmghho77@gmail.com](mailto:mmghho77@gmail.com)



<https://github.com/mmovasaghi>



<https://www.linkedin.com/in/mmovasaghi>

# Introduction

- Your major ?
- Your experiences ?
- Your Goal ?

Example:  
Computer Science – C#, php, C++

# Rules:

- Any question may be the best question in the world !
- Everyone must practice !

# Tittles

- Introduction
- What is Programming languages? And Who is the Programmer?
- What is Program?
- The Best Programming Language
- What is OOP?
- The different of OOP and functional programming? Which is better?
- Starting with Visual Studio 2019 or Visual Studio Code.

# Tittles cont.

- Introduction to C#
  - Hello World App
  - Data Types
  - Condition Statements (if, else, switch)
  - Loop Statements (for, while, do...while)
  - Method declaration (Function Signature)
  - Class declaration
    - What is OOP again!
    - Private vs. public
    - Constructor
    - Properties
    - Methods
  - Try/Catch Statement

# What is Programming languages?

- Programming is instructing your computer to do something. It could be as simple as printing your name to as complex as writing a program (AI).

Programming = **Mathematics** + **Creativity** + Syntax

# Which Programming Language is the best ?





# THE LORD OF THE RINGS ANALOGY TO PROGRAMMING LANGUAGES



**Python**

The Ent

DIFFICULTY  
★☆☆☆☆



Help little Hobbits (beginners) to understand programming concepts

Help Wizards (computer scientists) to conduct researches

Widely regarded as the best programming language for beginners

Easiest to learn

Widely used in scientific, technical & academic field, i.e. Artificial Intelligence

You can build website using Django, a popular Python web framework.

POPULARITY  
★★★★★

USED TO BUILD  
YouTube, Instagram, Spotify

AVG. SALARY  
\$107,000



**Java**

Gandalf

DIFFICULTY  
★★★★★



Wants peace & works with everyone (portable)

Very popular on all platforms, OS, and devices due to its portability

One of the most in demand & highest paying programming languages

Slogan: write once, work everywhere

POPULARITY  
★★★★★

USED TO BUILD  
Gmail, Minecraft, Most Android Apps, Enterprise applications

AVG. SALARY  
\$102,000



**C**

One Ring

DIFFICULTY  
★★★★★



The power of C is known to them all

Everyone wants to get its Power

Lingua franca of programming language

One of the oldest and most widely used language in the world

Popular language for system and hardware programming

A subset of C++ except the little details

POPULARITY  
★★★★★

USED TO BUILD  
Operating systems and hardware

AVG. SALARY  
\$102,000



**C++**

Saruman

DIFFICULTY  
★★★★★



Everyone thinks that he is the good guy

But once you get to know him, you will realize he wants the power, not good deeds

Complex version of C with a lot more features

Widely used for developing games, industrial and performance-critical applications

Learning C++ is like learning how to manufacture, assemble, and drive a car

POPULARITY  
★★★★★

USED TO BUILD  
Operating systems, hardware, and browsers

AVG. SALARY  
\$104,000



**JavaScript**

Hobbit

DIFFICULTY  
★★★☆☆



Frequently underestimated (powerful)

Well-known for the slow, gentle life of the Shire (web browsers)

"Java and Javascript are similar like Car and Carpet are similar" - Greg Hewgill

Most popular clients-side web scripting language

A must learn for front-end web developer (HTML and CSS as well)

POPULARITY  
★★★★★

USED TO BUILD  
Paypal, front-end of majority websites

AVG. SALARY  
\$99,000



**C#**

Elf

DIFFICULTY  
★★★★★



Beautiful creature (language), used to stay in their land, Rivendell (Microsoft Platform), but recently started to open up to their neighbours (open source)

A popular choice for enterprise to create websites and Windows application using .NET framework

Can be used to build website with ASP.NET, a web framework from Microsoft

Similar to Java in basic syntax and some features

POPULARITY  
★★★★★

USED TO BUILD  
Enterprise and Windows applications

AVG. SALARY  
\$94,000



**Ruby**

Man (Middle Earth)

DIFFICULTY  
★★★☆☆



Very emotional creature

They (some Ruby developers) feel they are superior & need to rule the Middle Earth

Mostly known for its popular web framework, Ruby on Rails

Focuses on getting things done

Designed for fun and productive coding

POPULARITY  
★★★★★

USED TO BUILD  
Hulu, Groupon, Slideshare

AVG. SALARY  
\$107,000



**PHP**

Orc

DIFFICULTY  
★★★☆☆



Ugly guy (language) and doesn't respect the rules (inconsistent and unpredictable)

Big headache to those (developers) to manage them (codes)

Yet still dominates the Middle-earth (most popular web scripting language)

Suitable for building small and simple sites within a short time frame

Supported by almost every web hosting services with lower price

POPULARITY  
★★★★★

USED TO BUILD  
Wordpress, Wikipedia, Flickr

AVG. SALARY  
\$89,000



**Objective-C**

Smaug

DIFFICULTY  
★★★★★



Lonely and loves gold

Primary language used by Apple for Mac OS X & iOS

Choose this if you want to focus on developing iOS or OS X apps only

Consider to learn Swift (newly introduced by Apple in 2014) as your next language

POPULARITY  
★★★★★

USED TO BUILD  
Most iOS Apps and part of Mac OS X

AVG. SALARY  
\$107,000



# Programming Languages vs. Languages

# Start Coding ...

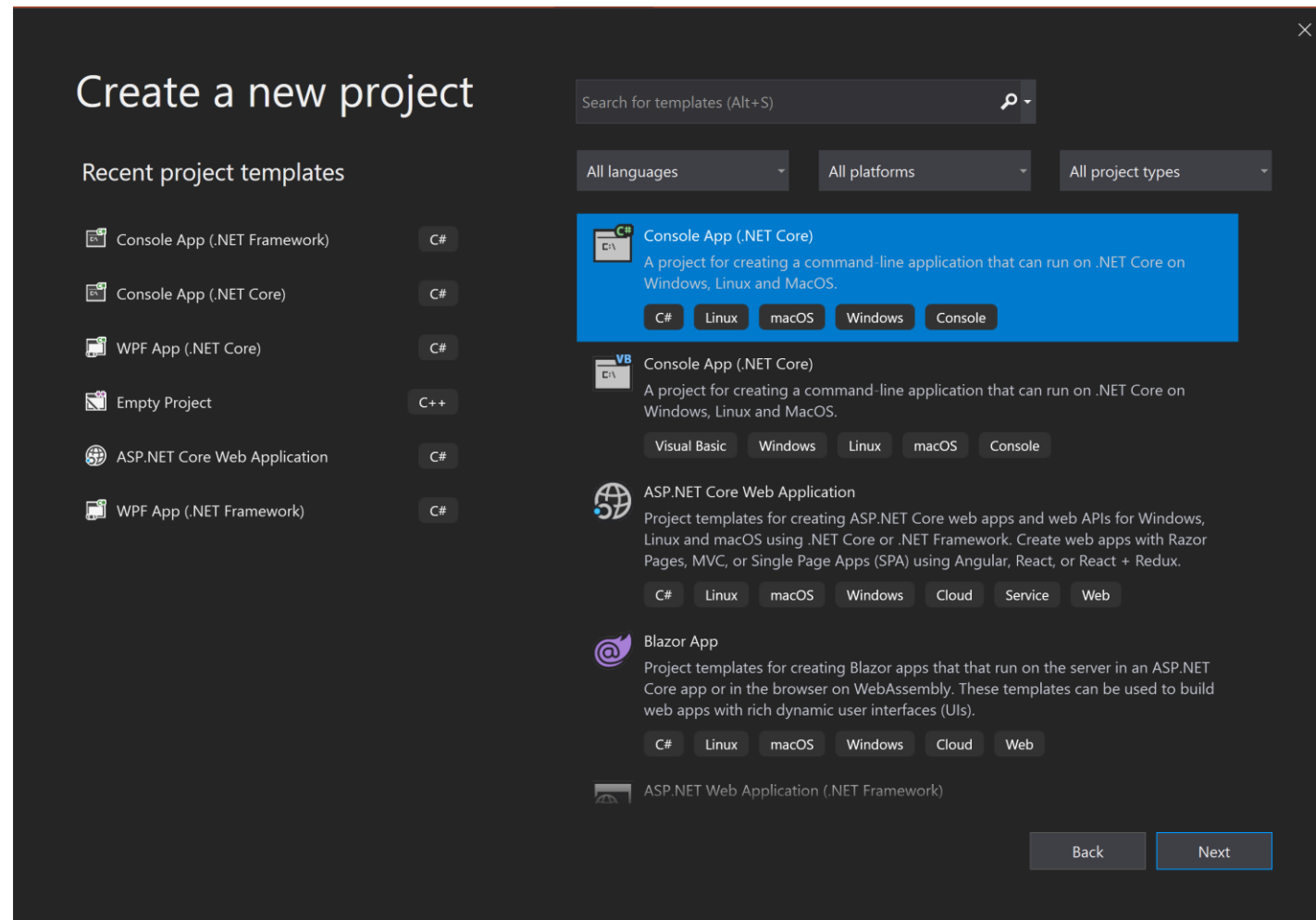
- Installing Visual Studio 2017 or 2019
  - Note: Add C# packages in to your install packages
- How !? Google it ... !!

**For Visual Studio Code**

<https://code.visualstudio.com/docs/editor/debugging>



# Create a Project on Visual Studio



# Hello World App

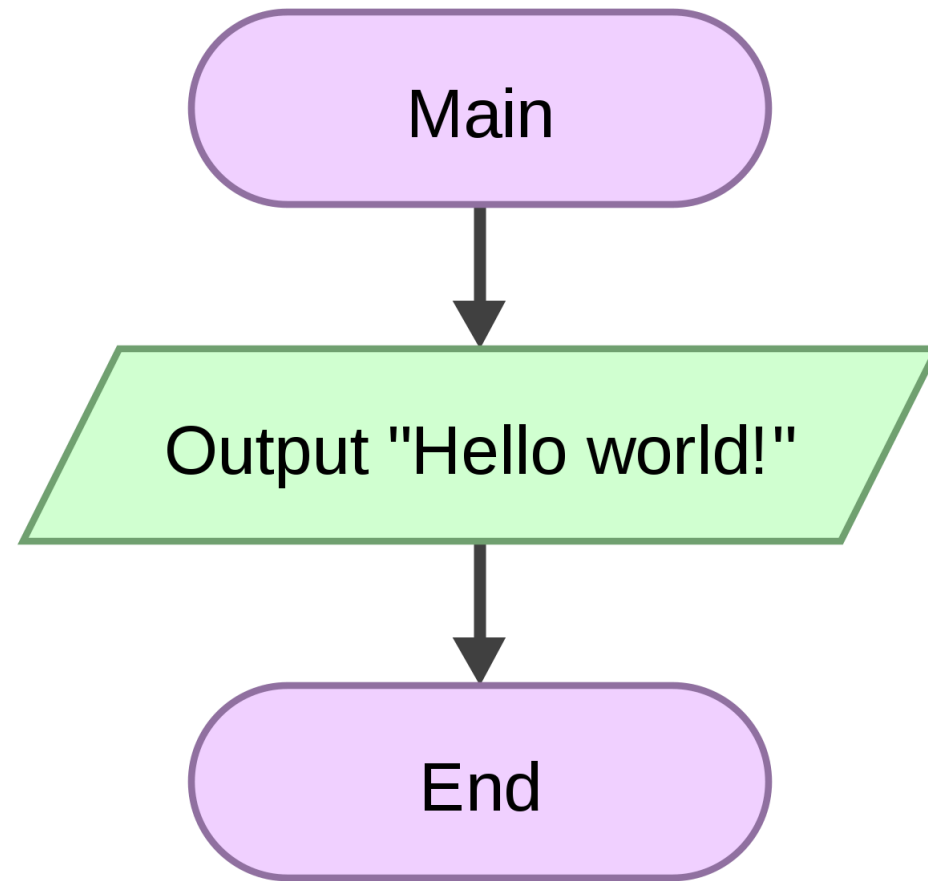
Code:

```
Console.WriteLine("Hello World!");
```

# Please Open the door !



# Flow Control





# Data Types

- String : "Hello World!"
- Char : 'H'
- Number :
  - int : 18
  - double : 123.387
  - Float : 182.233
  - decimal : 1928.7387
- Bool : true,false

# Data Types

```
int a = 20;
```

```
double b = 23.3928;
```

```
float c = 23.398;
```

```
decimal f = 234.239480m;
```

```
char d = 'H';
```

```
string e = "Hello World!";
```

# Get data and Print data

Code:

```
Console.WriteLine();  
Console.Write();  
Console.ReadLine();  
Console.Read();
```

# Homework 1

یک برنامه بنویسین که از کاربر به ترتیب موارد زیر رو بگیره :

اسم کوچک

فامیلی

سن

شماره تلفن همراه

```
Hello,  
Please Enter Your FirstName : mohamad  
Please Enter Your Last Name : hoseini  
Please Enter Your Age : 18  
Please Enter Your Phone : 0912984392  
-----  
Hello mohamad hoseini,  
You sed me that you are 18 years old now.  
And Your phone number is 0912984392
```

بعد از این که این ها رو از کاربر گرفت، یک خط تیره

زیر ورودی ها بکشه و همه این ها رو با یک متن باحال چاپ کنه.



# Converting

Code:

```
int a = int.Parse("12");  
double b = double.Parse("36.40");  
float c = float.Parse("12.87");  
decimal d = decimal.Parse("12.345");  
  
int e = Convert.ToInt32("123");  
double f = Convert.ToDouble("12.34532");  
decimal g = Convert.ToDecimal("12.345");
```

# Homework 2

یک برنامه بنویسین که از کاربر این اطلاعات رو بگیره :

اسم کاربر

سال فعلی

سال تولد کاربر

```
Hello,  
Please Enter Your Name : Mohamad  
Please Enter Your Present Year : 1398  
Please Enter Your Bith Year : 1362  
-----  
Hi Mohamad,  
You sed me that you are 36 years old now.
```

بعد از گرفتن این اطلاعات سن کاربر رو محاسبه کنه و به یک

شکل باحال چاپ کنه برای کاربر.



# Comment

// Write some thing in one line

/\*

Write some thing in more than one line

.....

.....

\*/

# Print data

## Code:

```
int a = 30;
double b = 40.2343;
string c = "THE C STRING";
Console.WriteLine($"Salam,\nnumber a = {a} " +
    $"and number b = {b} " +
    $"and the C string = {c}\n" +
    $"This is the First One.");
Console.WriteLine("-----");
Console.WriteLine("Salam,\nnumber a = {0} " +
    "and number b = {1} " +
    "and the C string = {2}\n" +
    "This is the Second One.", a, b, c);
```

```
Salam,
number a = 30 and number b = 40.2343 and the C string = THE C STRING
This is the First One.
-----
Salam,
number a = 30 and number b = 40.2343 and the C string = THE C STRING
This is the Second One.
```



# String Format

## Code:

```
int a = 30;
double b = 40.2343;
string c = "THE C STRING";
string First = "";
string Second = "";
First = string.Format($"Salam,\nnumber a = {a} " +
    $"and number b = {b} " +
    $"and the C string = {c}\n" +
    $"This is the First One.");
Second = string.Format("Salam,\nnumber a = {0} " +
    "and number b = {1} " +
    "and the C string = {2}\n" +
    "This is the Second One.", a, b, c);
Console.WriteLine($"String First : {First}");
Console.WriteLine("-----");
Console.WriteLine($"String Second : {Second}");
```

```
String First : Salam,
number a = 30 and number b = 40.2343 and the C string = THE C STRING
This is the First One.
-----
String Second : Salam,
number a = 30 and number b = 40.2343 and the C string = THE C STRING
This is the Second One.
```

# Boolean algebra

X	Y	X OR Y
0	0	0
0	1	1
1	0	1
1	1	1

X	Y	X AND Y
0	0	0
0	1	0
1	0	0
1	1	1

1 == true

0 == false

# Boolean algebra

In the programming we have & and &&

The & is for **Bitwise-AND**.

The && is for **Expression-AND**.

In the programming we have | and ||

The | is for **Bitwise-OR**.

The || is for **Expression-OR**.

# Bitwise Boolean algebra ex.

1 **&** 2 => (01) **&** (10) = (00) => false

0 **&** 3 => (00) **&** (11) = (00) => false

1 **&** 3 => (01) **&** (11) = (01) => true

1 **|** 2 => (01) **|** (10) = (11) => true

0 **|** 3 => (00) **|** (11) = (11) => true

0 **|** 0 => (00) **|** (00) = (00) => false

# Expression Boolean algebra ex.

1 && 2 => true

0 && 3 => false

0 && 2 && 1 => false

true && 2 => true

false && false && true => false

true && true => true

false && false => false

0 && 0 => false

1 || 2 => true

0 || 3 => true

0 || 2 || 1 => true

true || 2 => true

false || false || true => true

true || true => true

false || false => false

0 || 0 => false

# If & else Statement

```
if (expression)
{
    do some thing
}
else
{
    do some thing else
}
```

# If & else Statement cont.

```
if (expression1)
{
    do some thing
}
else if (expression2)
{
    do some thing else
}
else if (expression3)
{
    do some thing else
}
else
{
    do some thing else
}
```

# Homework 3

برنامه ای بنویسید که 3 عدد از کاربر بگیرد، بزرگ ترین آن را چاپ کند.





# Homework 4

برنامه ای بنویسید که 3 عدد از کاربر بگیرد، کوچکترین آن را چاپ کند.



# Homework 5

برنامه ای بنویسید که 3 عدد از کاربر بگیرد، از بزرگ به کوچک مرتب کرده و آن ها را چاپ کند.



# Switch...Case Statement

```
switch (expression)
{
    case 1:
        do some thing
        break;
    case 2:
        do some thing
        break;
    case 12:
        do some thing
        break;
    default:
        do some thing
        break;
}
```

```
switch (expression)
{
    case "hello":
        do some thing
        break;
    case "hi":
        do some thing
        break;
    case "salam":
        do some thing
        break;
    default:
        do some thing
        break;
}
```

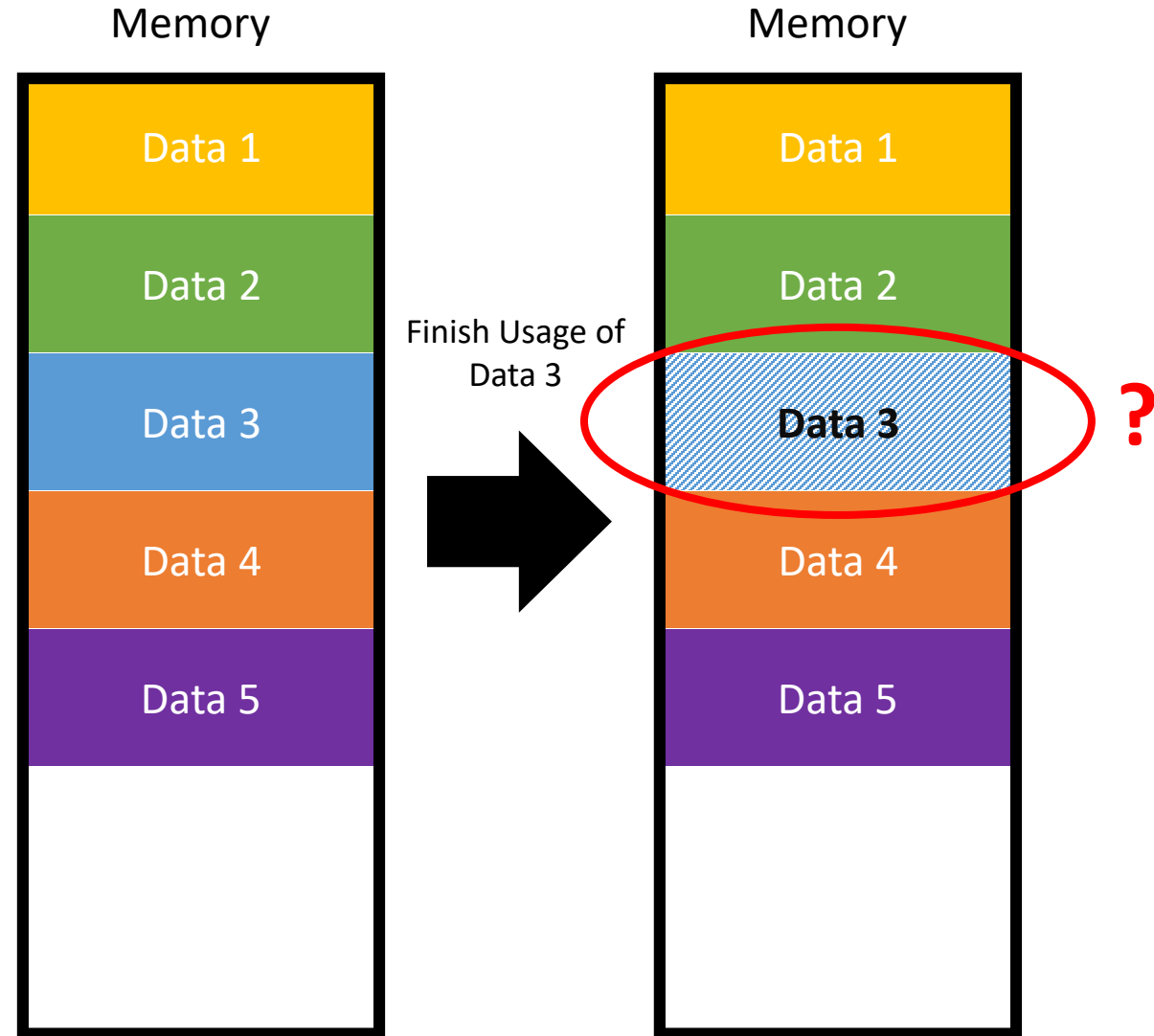
# Loop Statement

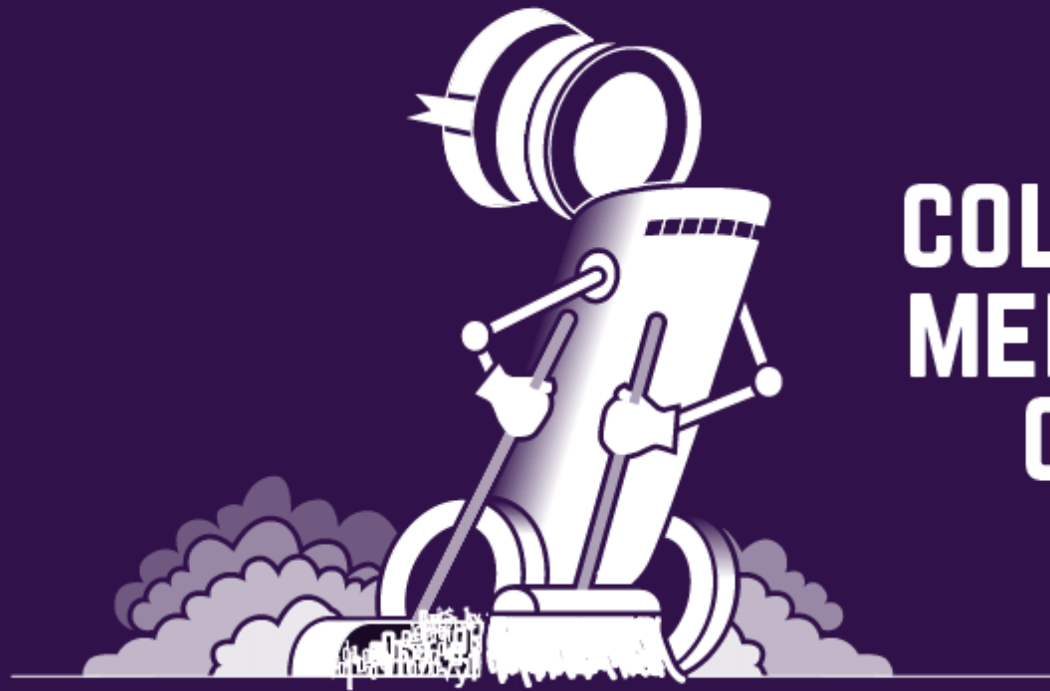
```
while (expression)
{
    do some thing
}
```

```
do
{
} while (expression);
```

```
for (initial ;expression ;do some thing)
{
}
```

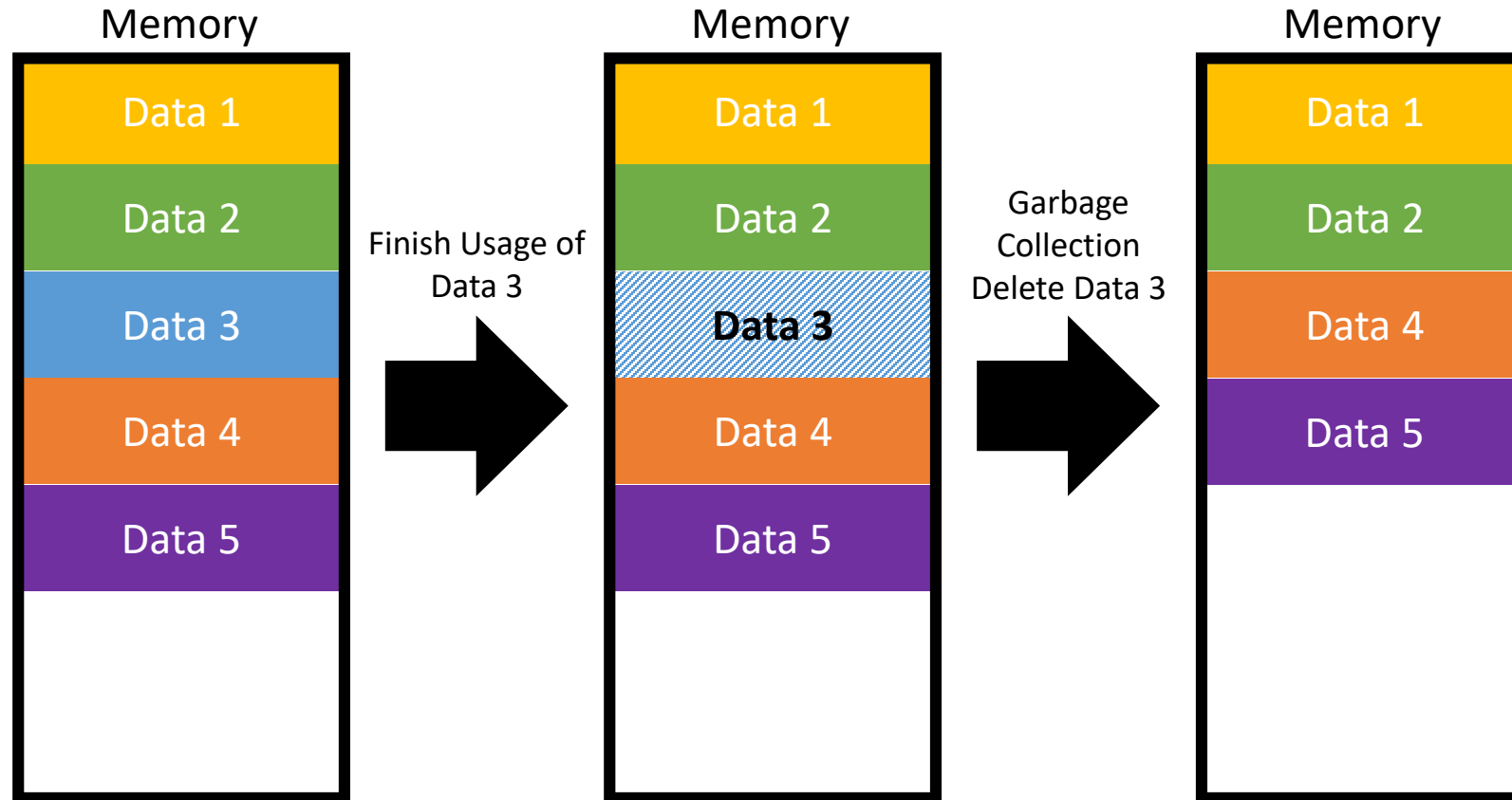
# Memory Model





# GARBAGE COLLECTION, A MEMORANDUM ON MEMORY IN C#

# Garbage Collection



# StringBuilder

Code:

```
StringBuilder Sample = new StringBuilder();
```



# Homework 6

برنامه ای بنویسید که 10 عدد از کاربر بگیرد، بزرگترین و کوچک ترین آن را چاپ کند.



# Homework 7

برنامه ای بنویسید که از کاربر نام او را گرفته و سپس آن را اسپیل کند، مانند مثال:

Input:

Hoseini

Output:

H-o-s-e-i-n-i



# Random Number

```
Random random = new Random();  
int num = random.Next(1, 1000);
```

# Homework 8

```
Enter an integer number : 500
My number is more than 500
Enter an integer number : 750
My number is more than 750
Enter an integer number : 875
My number is less than 875
Enter an integer number : 800
My number is less than 800
Enter an integer number : 775
My number is more than 775
Enter an integer number : 785
My number is less than 785
Enter an integer number : 780
My number is less than 780
Enter an integer number : 777
My number is less than 777
Enter an integer number : 776
That's correct, the number is 776
```



برنامه ای بنویسید که یک عدد رندم بسازد بین 1 تا 1000 سپس از کاربر ورودی بگیرد، اگر عددی که کاربر وارد کرد بزرگ تر از عددی بود که کامپیوتر در نظر گرفته بود بگوید عدد مورد نظر کوچک تر است، اگر مساوی بود تبریک بگوید و برنامه تمام شود، اگر کوچکتر بود بگوید عدد مورد نظر بزرگ تر است و همین منوال را ادامه دهد تا عدد مورد نظر را کاربر بیابد.

# Binary Search

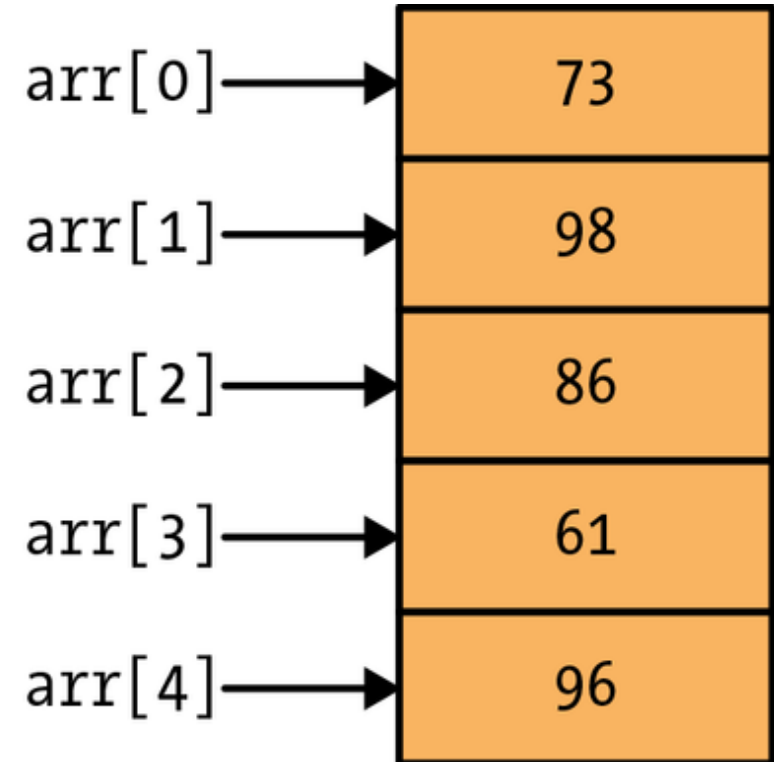
Hack

Search for 47

0	4	7	10	14	23	45	47	53
---	---	---	----	----	----	----	----	----

# Array

[0]	[1]	[2]	[3]	[4]
73	98	86	61	96



# Array

```
int[] a = new int[10];
```

```
double[] b = new double[10];
```

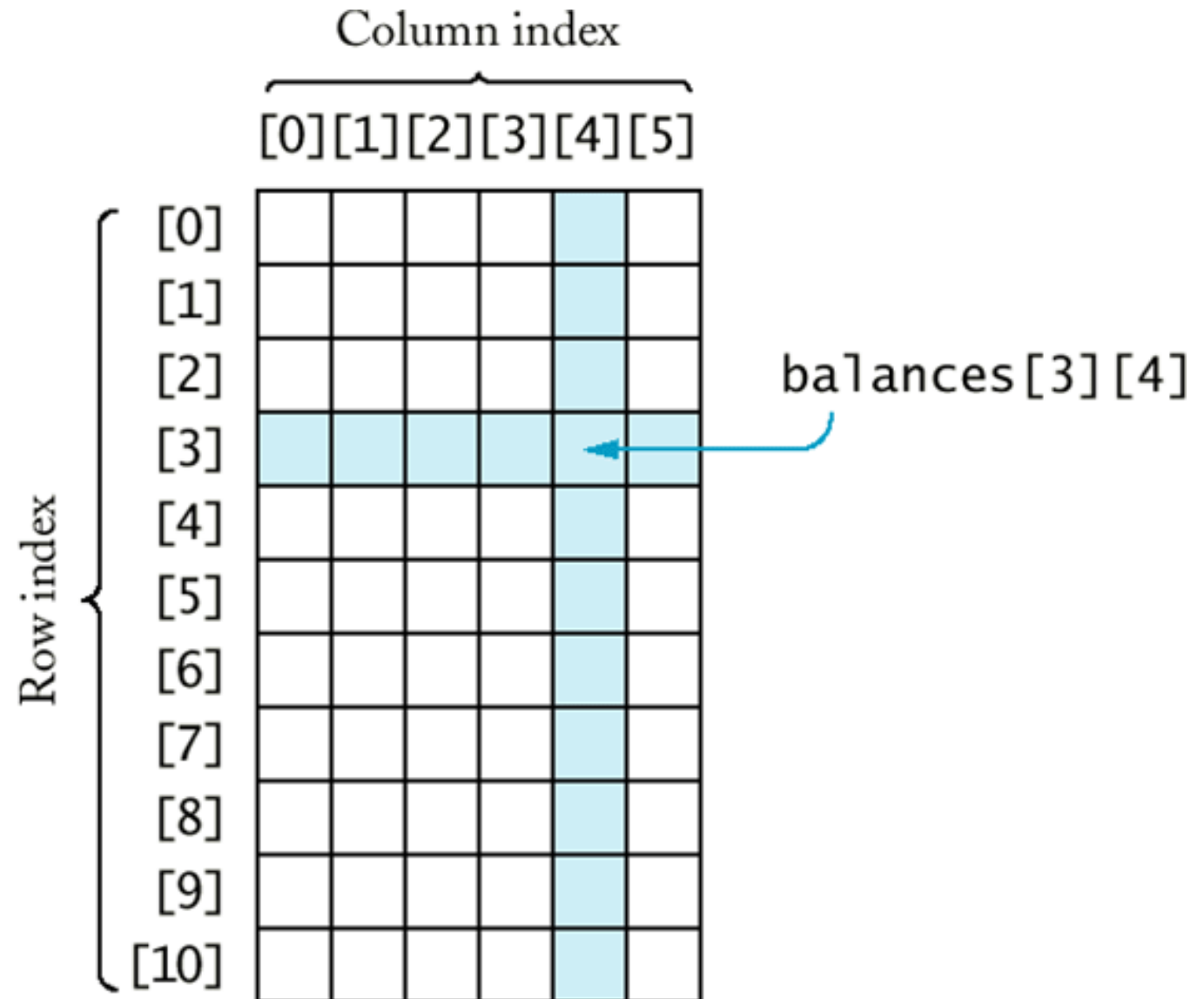
```
float[] c = new float[10];
```

```
decimal[] f = new decimal[10];
```

```
char[] d = new char[10];
```

```
string[] e = new string[10];
```

# 2D Array





# 2D Array

```
int[,] a = new int[10, 20];
```

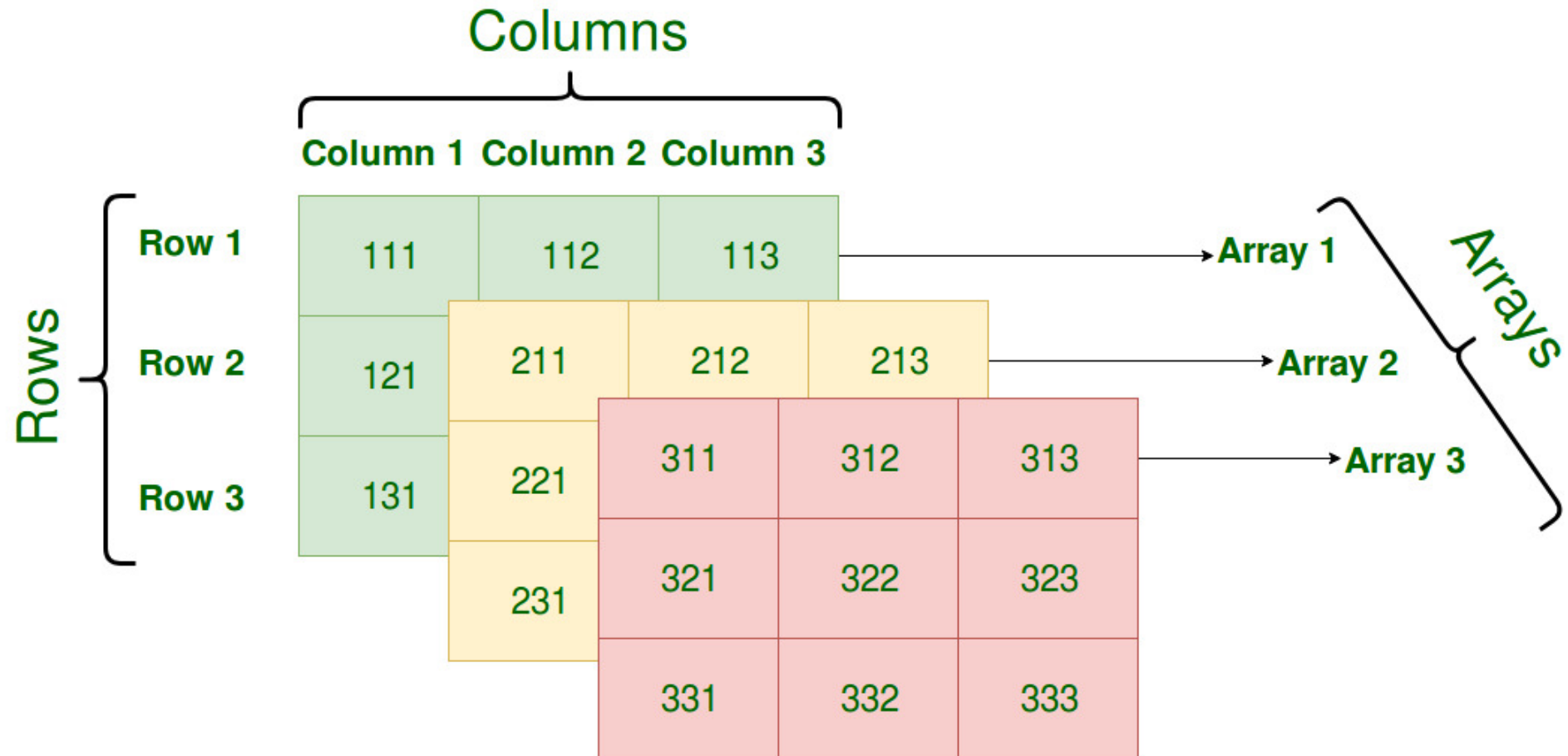
```
double[,] b = new double[10, 20];
```

```
float[,] c = new float[10, 20];
```

```
decimal[,] f = new decimal[10, 20];
```

```
char[,] d = new char[10, 20];
```

# 3D Array



# 3D Array

```
int[, ,] a = new int[10, 20, 30];
```

```
double[, ,] b = new double[10, 20, 30];
```

```
float[, ,] c = new float[10, 20, 30];
```

```
decimal[, ,] f = new decimal[10, 20, 30];
```

```
char[, ,] d = new char[10, 20, 30];
```

```
string[,] e = new string[10, 20];
```

# Homework 8

برنامه ای بنویسید که 10 عدد از کاربر بگیرد، بزرگترین و کوچک ترین آن را چاپ کند.  
سپس از کاربر به طور متوالی عدد بگیرد و اعلام کند که عدد وارد شده آیا در بین آن 10 عدد می باشد یا نه.



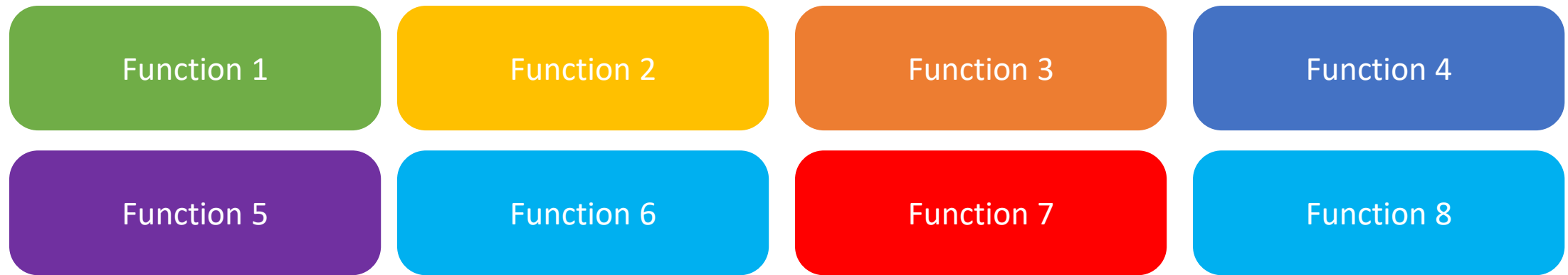
# Functional Programming

# Methods (or Functions)

$$y = F(x)$$

# Methods (or Functions)

$$y = F(x, z, w)$$



Main code {

some other code

Function 6

some other code

Function 7

Function 2

some other code

Function 1

}



# Methods

```
Return-Type Method-Name(Params1, Params2, ...)  
{  
    do some thing  
    return some-thing;  
}
```

# Methods

## Examples:

- Print Method
- Sum & print Method
- Get & sum & print Method
- Parameter Passing
- Return
- Sum & multiply & ... Methods
- Get number from user and do some thing with methods

# Homework 9

برنامه ای بنویسید که از کاربر ابتدا نام و نام خانوادگی او را بگیرد و چک کند که نام او شامل اعداد نباشد، در صورتی که شامل عدد بود به کاربر اعلام کند که ورودی شما باید فقط شامل حروف باشد و دوباره ورودی را بگیرد، این کار را تا موقعی که کاربر ورودی صحیح را وارد نکرده است ادامه دهد.

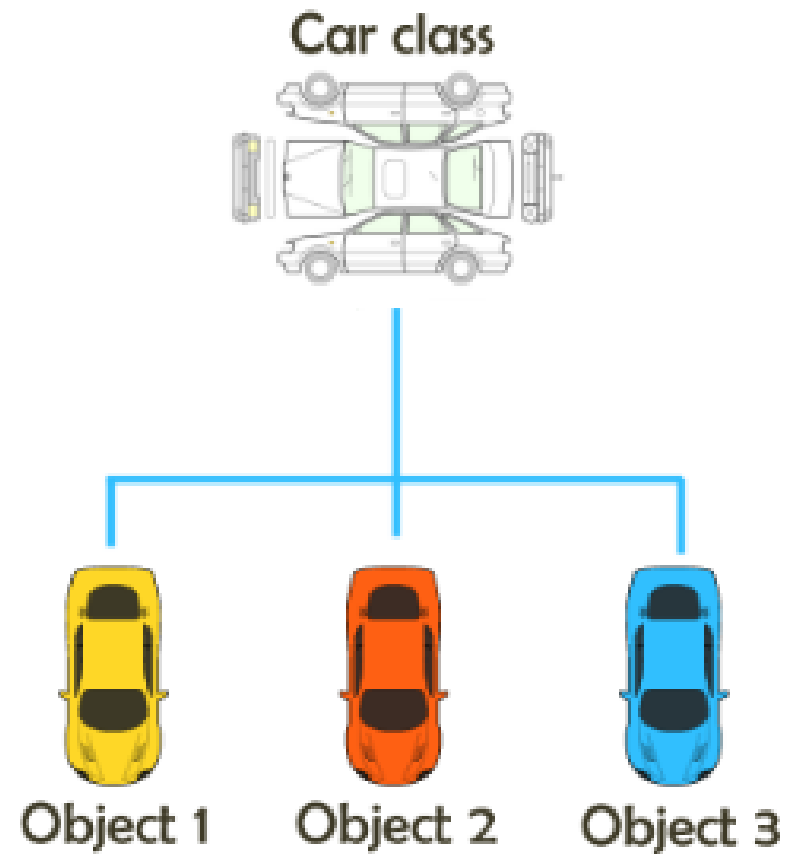
پس از این به کاربر اعلام کند که شماره تلفن خود را وارد کند، برای شماره تلفن هم ابتدا چک کند که شماره تلفن دقیقا 11 رقم و تنها شامل عدد باشد، در غیر این صورت به کاربر اعلام کند که ورودی نامعتبر است و باید دوباره شماره تلفن خود را وارد کند.

پس از گرفتن این سه مورد، این پارامتر ها را برای تابع نمایش روی صفحه ارسال کند تا آن تابع خروجی را به شکل یک جمله چاپ کند.

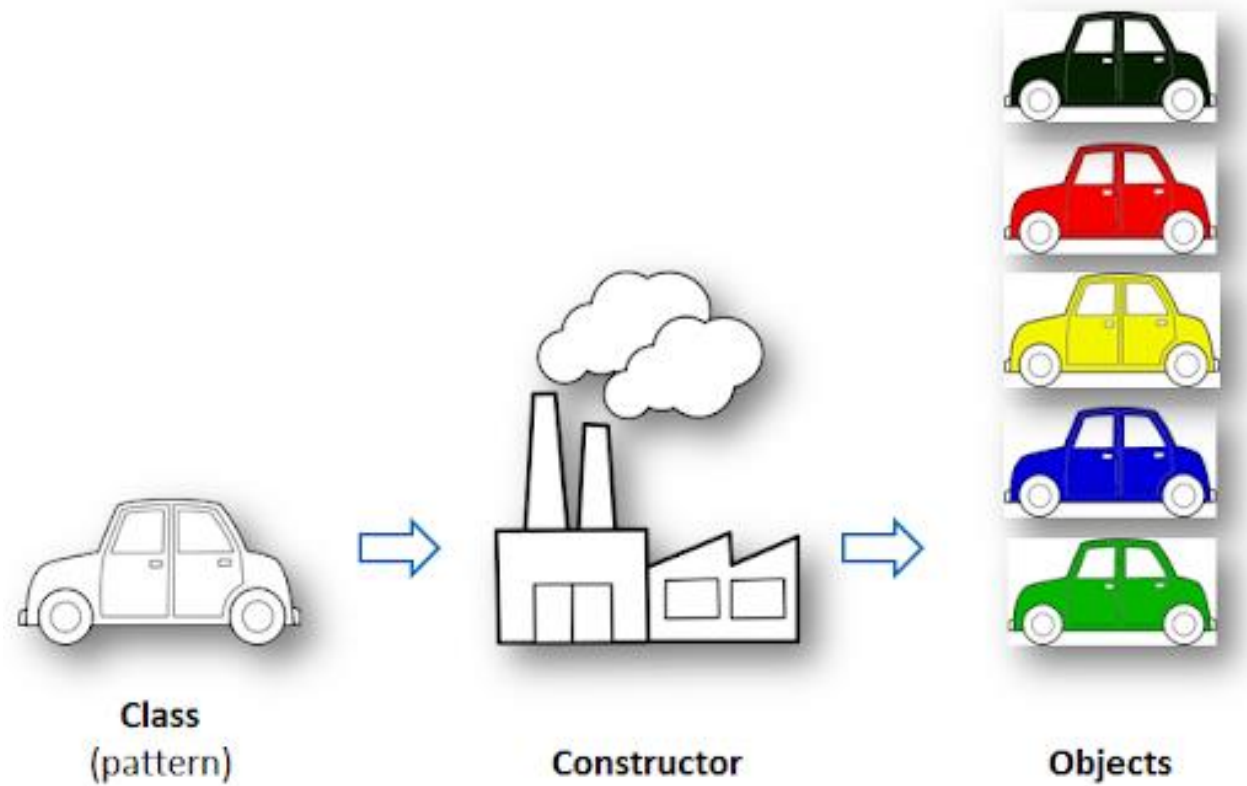
# OOP

# Object Oriented Programming

# OOP



# OOP



# OOP

Code:

```
class Car
{
    public void OpenDoorFromOutside()
    {

    }
    private void OpenDoorFromInside()
    {

    }
}
```

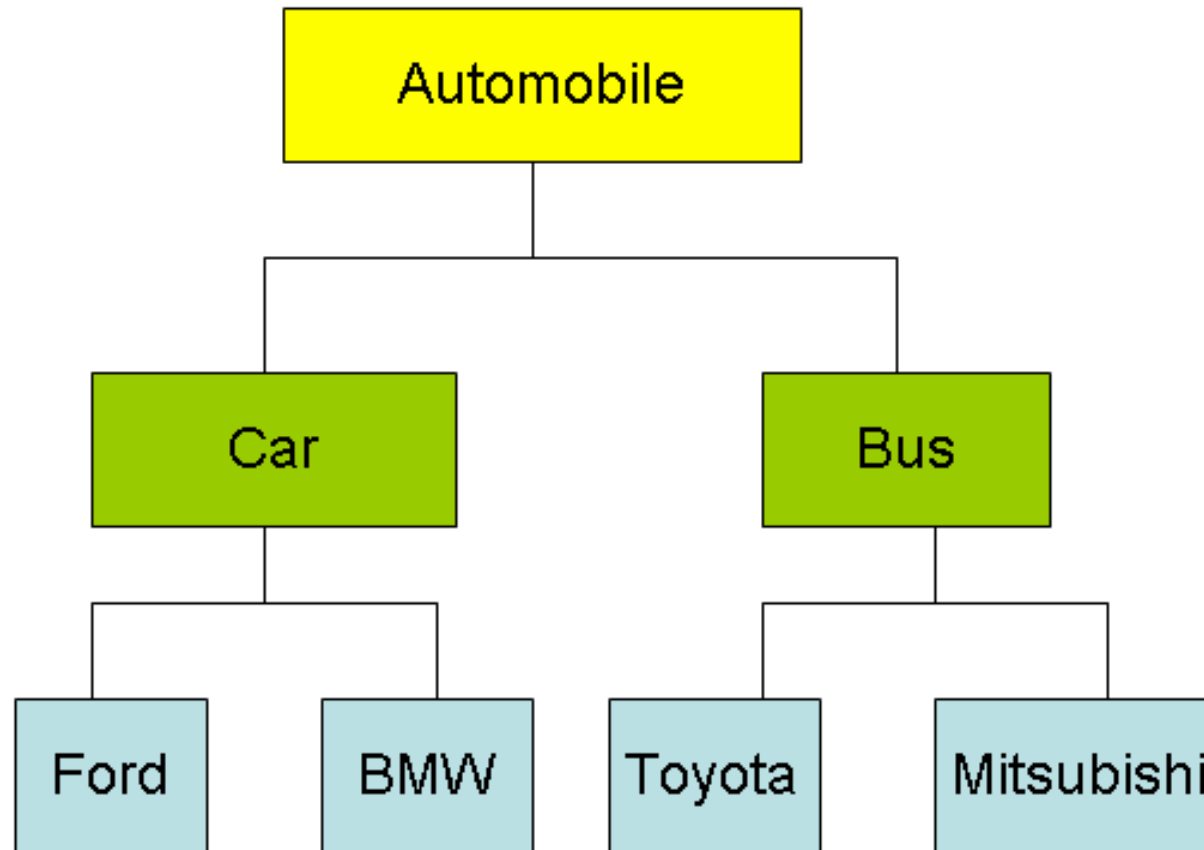
# OOP

- Methods (Static, dynamic)
- Properties
  - Automatic
  - Manual
- Use Object
- Constructor
- Public & Private
- Inheritance\*
- Protected props and methods\*
- Interfaces\* and Abstract-Class\*



# Functional Programming vs. Object Oriented Programming

# Inheritance\*



# File Processing

*CRUD : Create, Read, Update, Delete*

## File Operations:

- Create
- Open
- Delete
- Write
- Read
- Append
- Check Existence
- ...

# File Processing (Create & Write)

Code:

```
string path = @"./MyTest.txt";
if (!File.Exists(path))
{
    // Create a file to write to.
    using (StreamWriter sw = File.CreateText(path))
    {
        sw.WriteLine("Hello");
        sw.WriteLine("And");
        sw.WriteLine("Welcome");
    }
}
```

# File Processing (Read)

Code:

```
string path = @"./MyTest.txt";  
// Open the file to read from.  
using (StreamReader sr = File.OpenText(path))  
{  
    string s = "";  
    while ((s = sr.ReadLine()) != null)  
    {  
        Console.WriteLine(s);  
    }  
}
```

# File Processing (Append)

Code:

```
string path = @".\\MyTest.txt";  
using (StreamWriter sw = File.AppendText(path))  
{  
    sw.WriteLine("This");  
    sw.WriteLine("is Extra");  
    sw.WriteLine("Text");  
}
```

# File Processing (Delete)

Code:

```
string path = @"./MyTest.txt";  
if (File.Exists(path))  
{  
    File.Delete(path);  
    Console.WriteLine("Deleted Successfully");  
}  
else  
{  
    Console.WriteLine("That path not exists.");  
}
```

# Binary File Processing

**Code:**

<https://github.com/MMovasaghi/BinaryFileProcessing>



# File Processing (Create Directory)

## Code:

```
// Specify the directory you want to manipulate.
string path = @"./MyDir";
// Determine whether the directory exists.
if (Directory.Exists(path))
{
    Console.WriteLine("That path exists already.");
    return;
}
// Try to create the directory.
DirectoryInfo di = Directory.CreateDirectory(path);
Console.WriteLine("The directory was created successfully at {0}.",Directory.GetCreationTime(path));
```

# File Processing (Delete Directory)

Code:

```
string subPath = @"./NewDirectory/NewSubDirectory";  
Directory.CreateDirectory(subPath);  
Directory.Delete(subPath);  
bool directoryExists = Directory.Exists(@"./NewDirectory");  
bool subDirectoryExists = Directory.Exists(subPath);  
Console.WriteLine("top-level directory exists: " + directoryExists);  
Console.WriteLine("sub-directory exists: " + subDirectoryExists);
```

# File Processing

For More Information check Microsoft docs for **System.IO**:

<https://docs.microsoft.com/en-us/dotnet/api/system.io?view=netframework-4.8>

# Exceptions

- Try/Catch/Finally
- Throw Exception
- New Exception