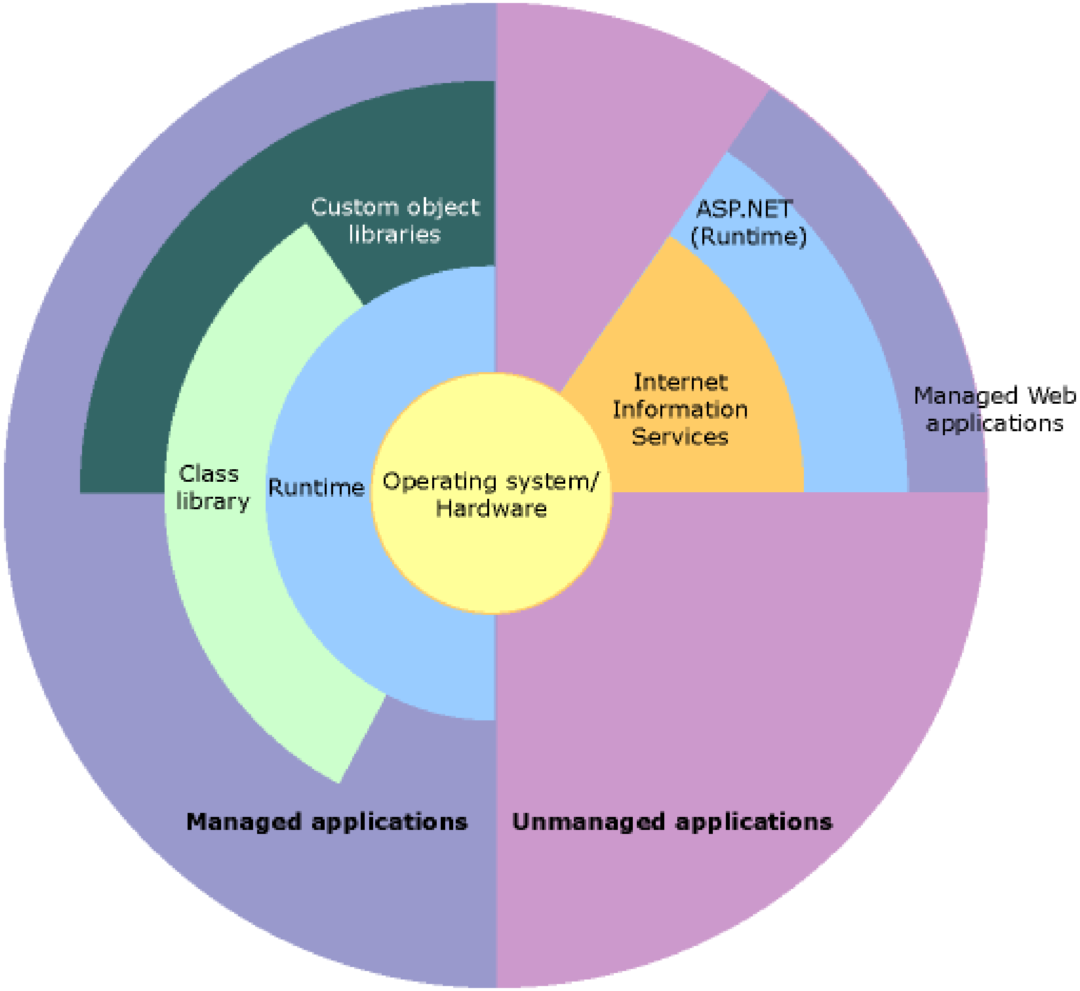
# Introduction To C#

Sushant Banerjee | sushantba@cybage.com | 7221

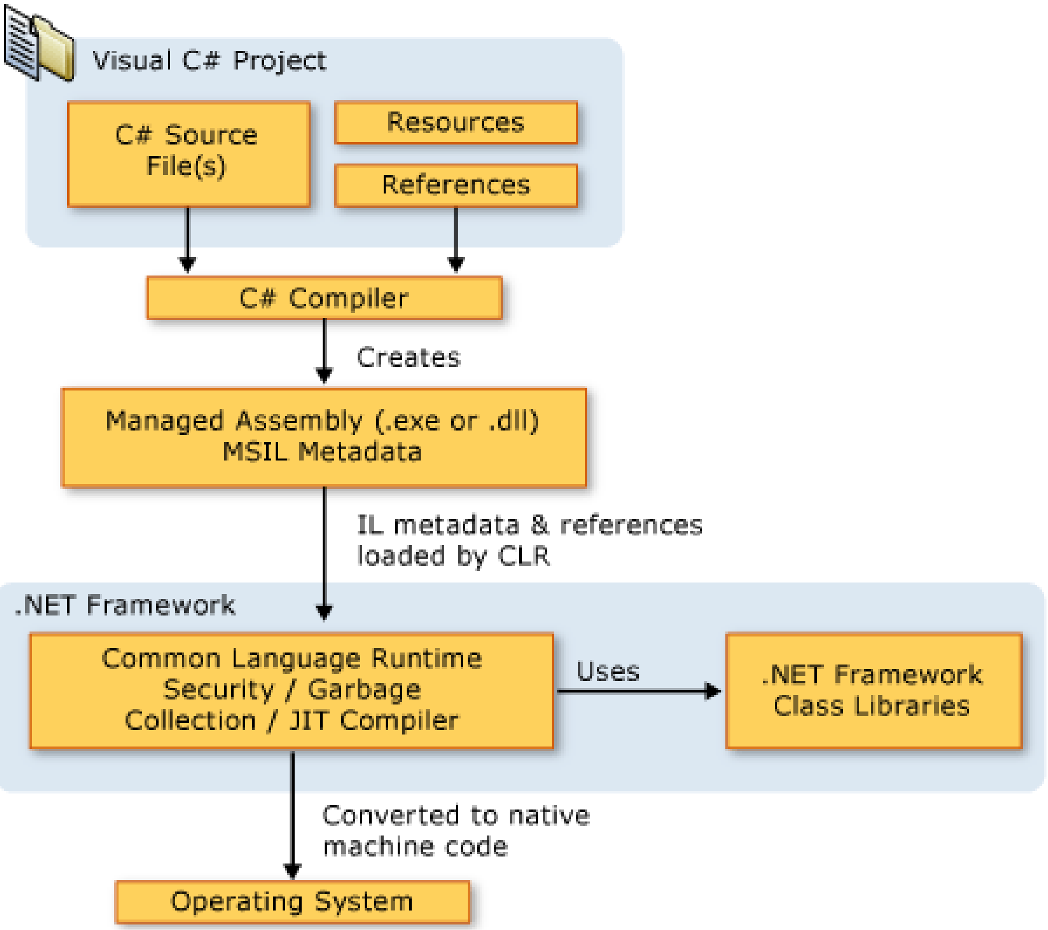
# Agenda

- Overview of .NET Framework
- Understanding Compilation Process
- Common Type System
- Primitive Types
- Classes and Objects
- Statements, Expressions and Operators
- Properties and Methods
- Access Modifiers
- Static classes and Static members
- Constructors and Destructors
- Arrays

# Overview of .NET Framework



# Managed Execution Process



Visual C# Project

C# Source File(s)

Resources

References

C# Compiler

Creates

Managed Assembly (.exe or .dll)
MSIL Metadata

IL metadata & references loaded by CLR

.NET Framework

Common Language Runtime
Security / Garbage Collection / JIT Compiler

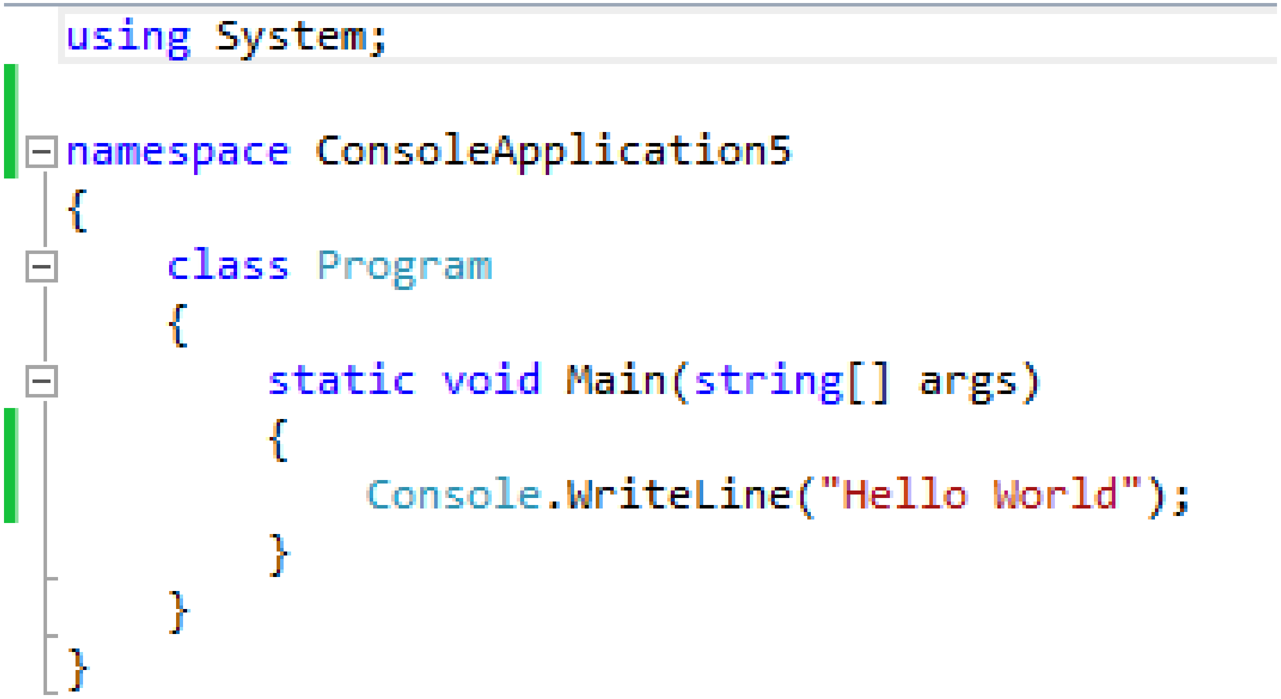
Uses

.NET Framework Class Libraries

Converted to native machine code

Operating System

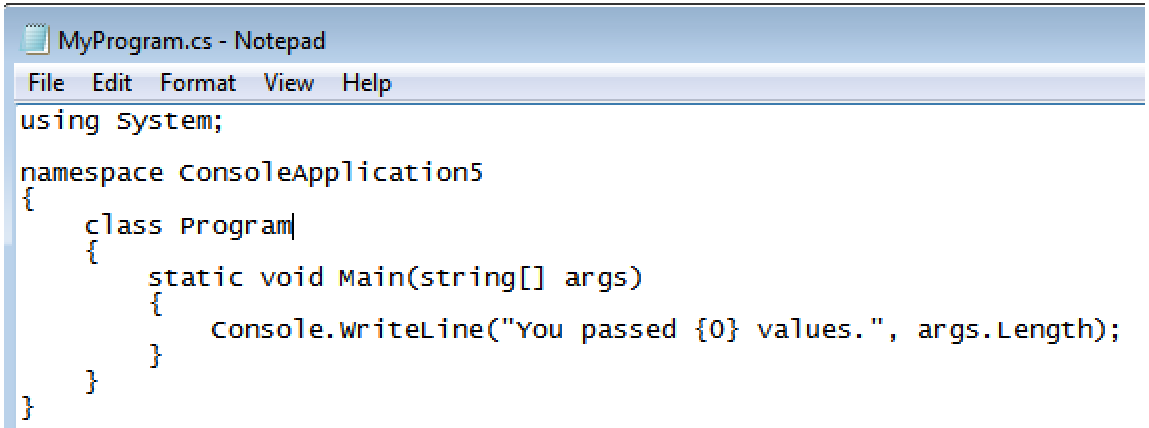
# Basic Structure of a C# Program



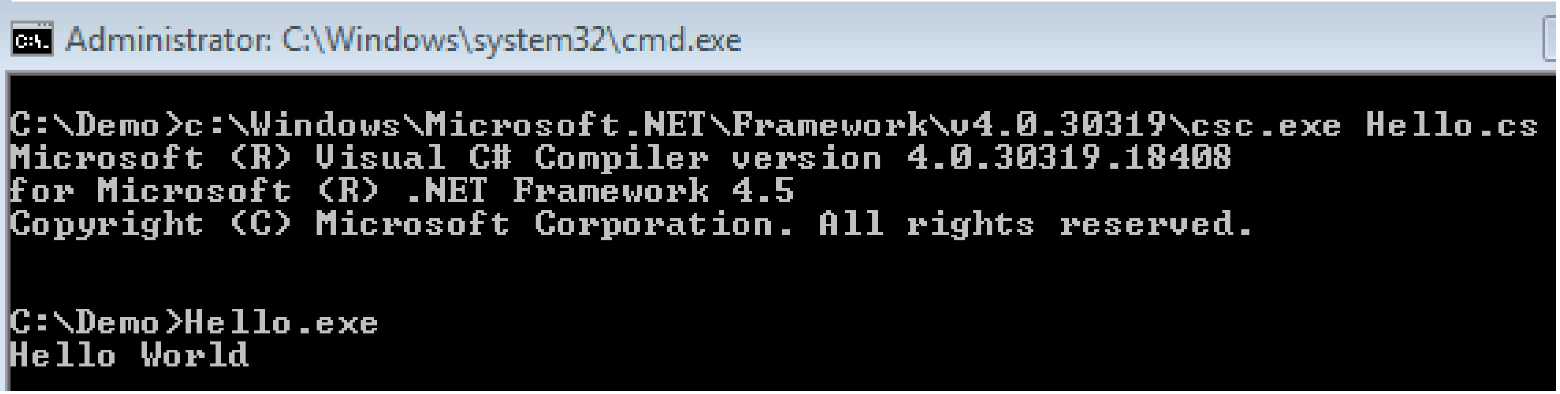
```
using System;

namespace ConsoleApplication5
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World");
        }
    }
}
```

# Compilation and Execution

- Compiling program
  - Using Command Prompt
  - Passing command line arguments.



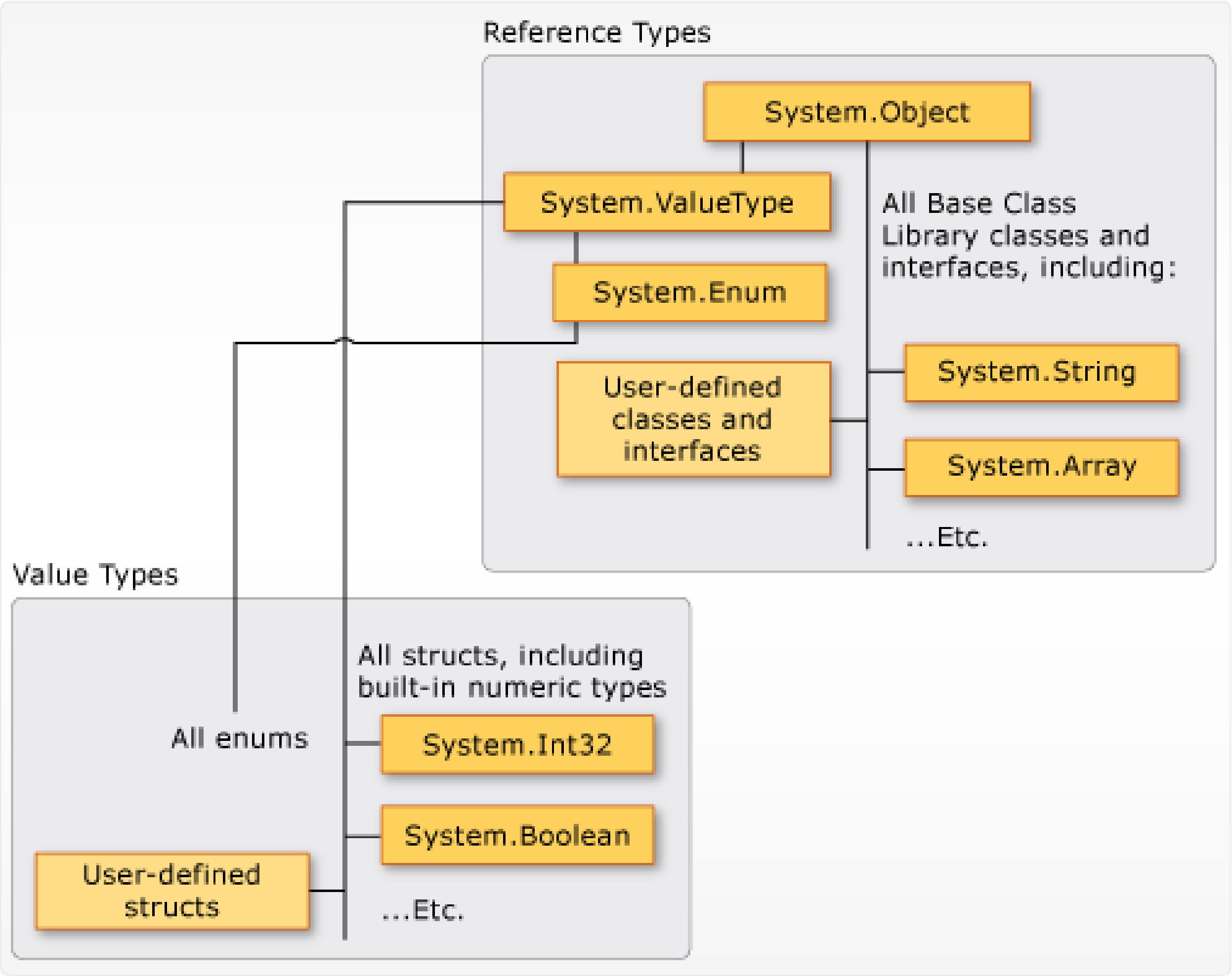
```
using System;

namespace ConsoleApplication5
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("You passed {0} values.", args.Length);
        }
    }
}
```



```
C:\Demo>c:\Windows\Microsoft.NET\Framework\v4.0.30319\csc.exe Hello.cs
Microsoft (R) Visual C# Compiler version 4.0.30319.18408
for Microsoft (R) .NET Framework 4.5
Copyright (C) Microsoft Corporation. All rights reserved.


C:\Demo>Hello.exe
Hello World
```

# Demo

# Common Type System (CTS)



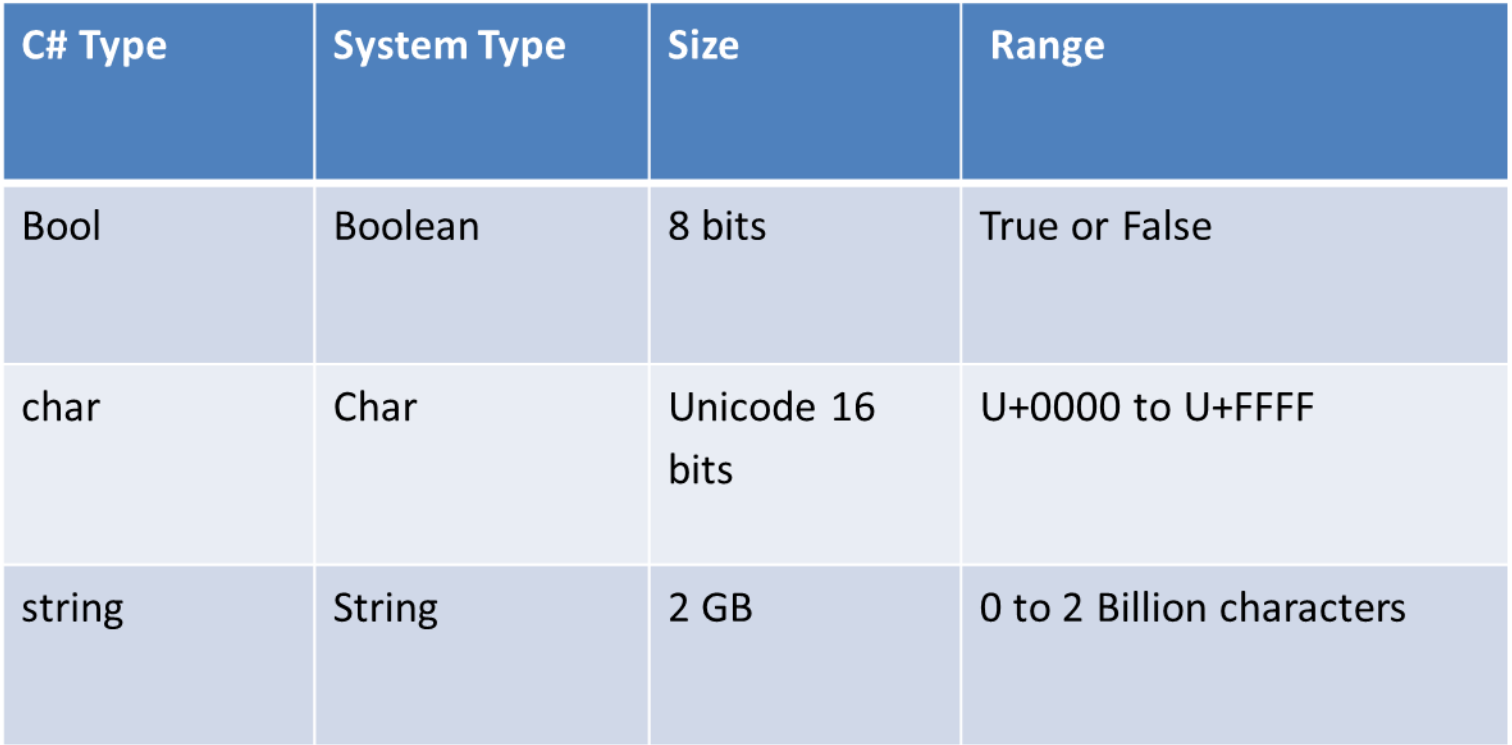
# Numeric Types - Integral

| C# Type | System Type | Suffix | Size | Range |
|---|---|---|---|---|
| sbyte | SByte | | 8 bits | -128 to 127 |
| short | Int16 | | 16 bits | -32,768 to 32,767 |
| int | Int32 | | 32 bits | -2,147,483,648 to 2,147,483,647 |
| long | Int64 | L | 64 bits | -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807 |

# Numeric Types – Real Numbers

| C# Type | System Type | Suffix | Size | Approximate Range | Precision |
|---|---|---|---|---|---|
| float | Single | F | 32 bits | ±1.5e−45 to ±3.4e38 | 7 digits |
| double | Double | D | 64 | ±5.0e−324 to ±1.7e308 | 15 – 16 digits |
| decimal | Decimal | M | 128 bits | ±1.0 × 10−28 to ±7.9 × 1028 | 28 – 29 digits |

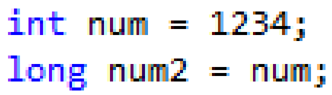
# Other Types



| C# Type | System Type | Size | Range |
|---|---|---|---|
| Bool | Boolean | 8 bits | True or False |
| char | Char | Unicode 16 bits | U+0000 to U+FFFF |
| string | String | 2 GB | 0 to 2 Billion characters |

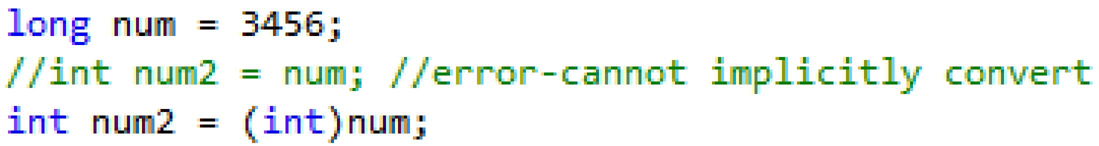
# Type Conversion

- Implicit Conversion



```
int num = 1234;
long num2 = num;
```

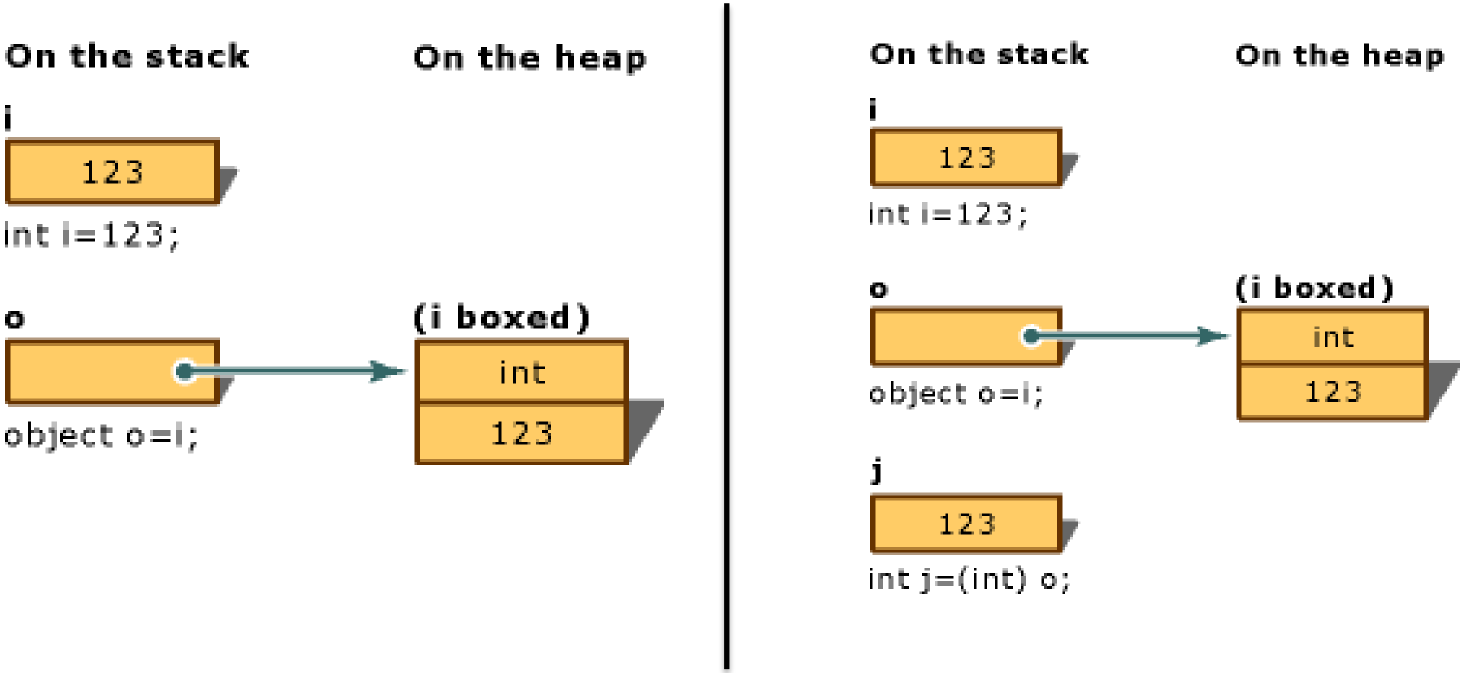
- Explicit Conversion



```
long num = 3456;
//int num2 = num; //error-cannot implicitly convert
int num2 = (int)num;
```

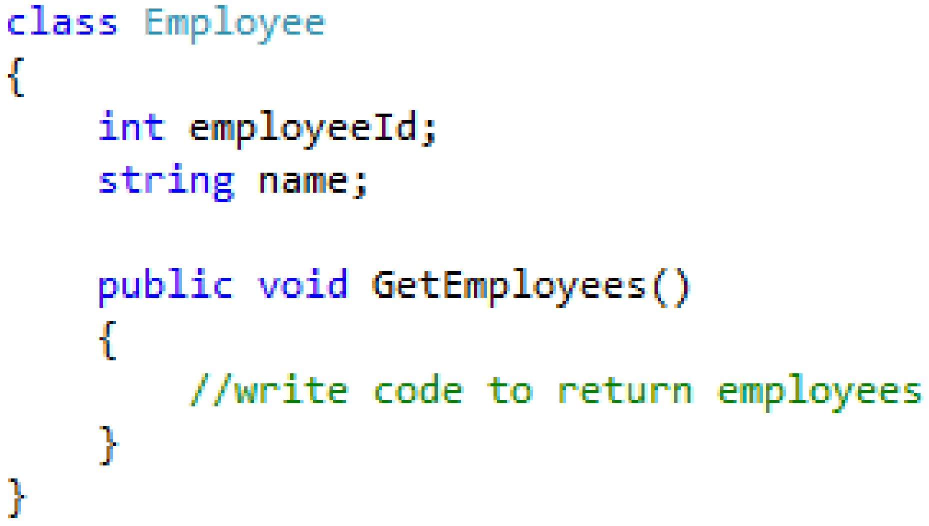
# Boxing and Unboxing

- Computationally expensive process



# Classes

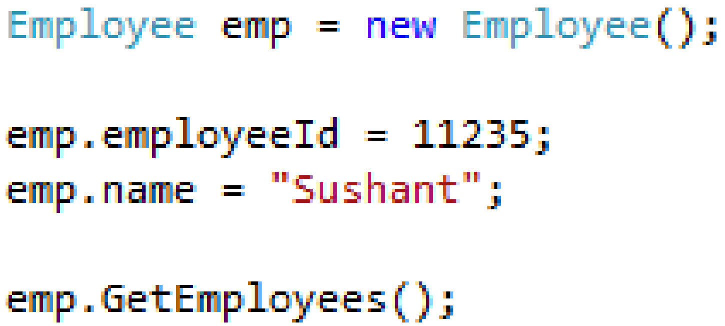
- A class is a custom reference type
- Contains members such as
  - Variables
  - Properties
  - Methods etc.



```
class Employee
{
    int employeeId;
    string name;

    public void GetEmployees()
    {
        //write code to return employees
    }
}
```

# Objects

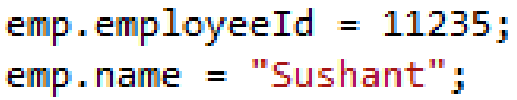
- An object is a block memory allocated based on the class
- Objects are created using “new” keyword
- An object is used to access members of the class.



```
Employee emp = new Employee();

emp.employeeId = 11235;
emp.name = "Sushant";

emp.GetEmployees();
```

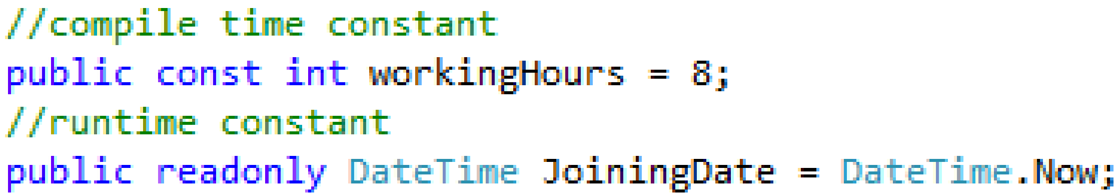
# Variables, Constants and Readonly

- Values of a variable may vary



```
emp.employeeId = 11235;
emp.name = "Sushant";
```

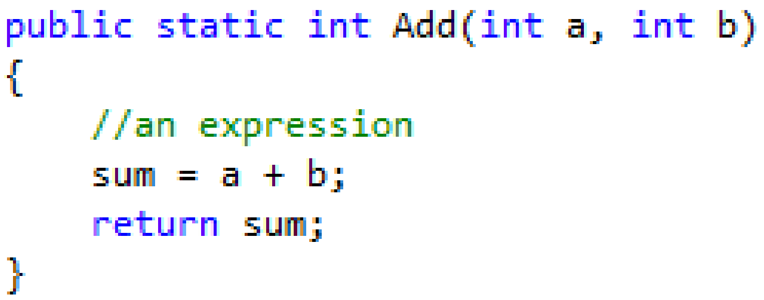
- Values of a constant is fixed



```
//compile time constant
public const int workingHours = 8;
//runtime constant
public readonly DateTime JoiningDate = DateTime.Now;
```

# Statements and Expressions

- Selection statements
  - If, else, switch, case
- Iteration statements
  - Do, for, foreach, in, while
- Jump statements
  - Break, continue, default, goto, return
- Expressions are sequence of one or more operands and operators
- Expressions can be evaluated to a single value.



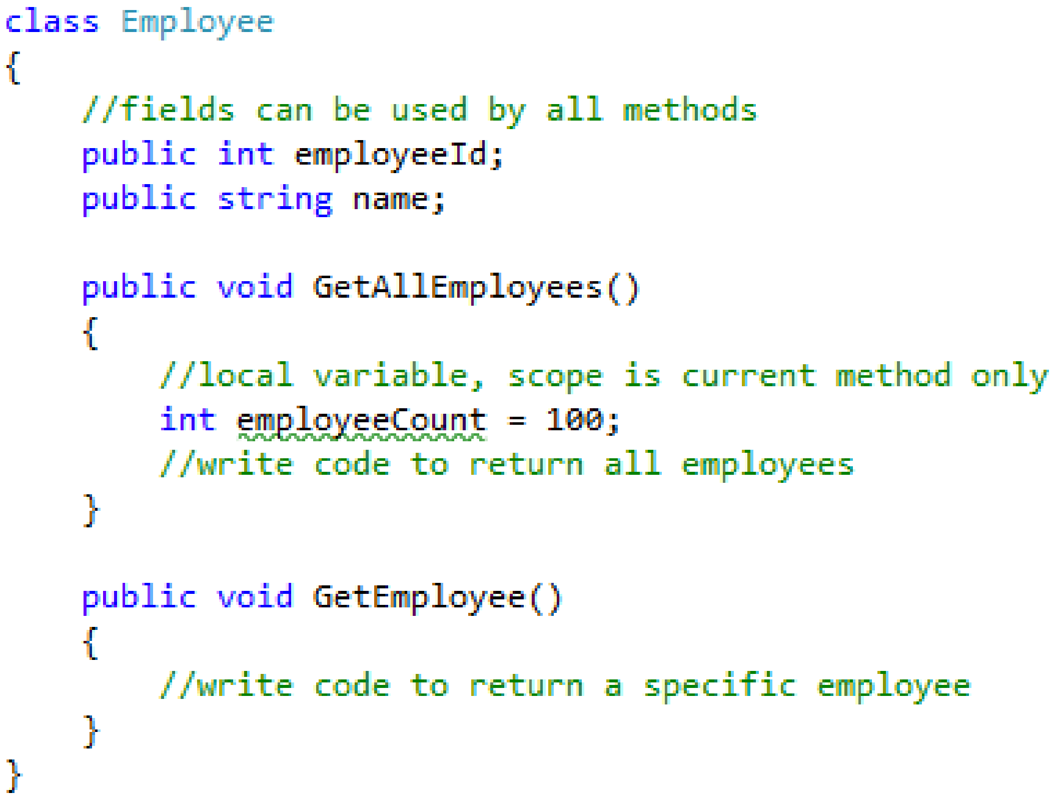
```
public static int Add(int a, int b)
{
    //an expression
    sum = a + b;
    return sum;
}
```

# Operators

- An operator is used along with operands to create expressions
- Unary operators
    - X++, X--, ++X, --X
- Binary operators or Arithmetic operators
    - X + Y, X – Y, X * Y, X / Y, X % Y
- Relational or Comparison operators
    - X > Y, X < Y, X >= Y, X <= Y, X == Y, X != Y
- Conditional AND – X && Y
- Condition OR – X || Y

# A Field

- A field is a variable declared in class level
- A field initialized immediately before the constructor.

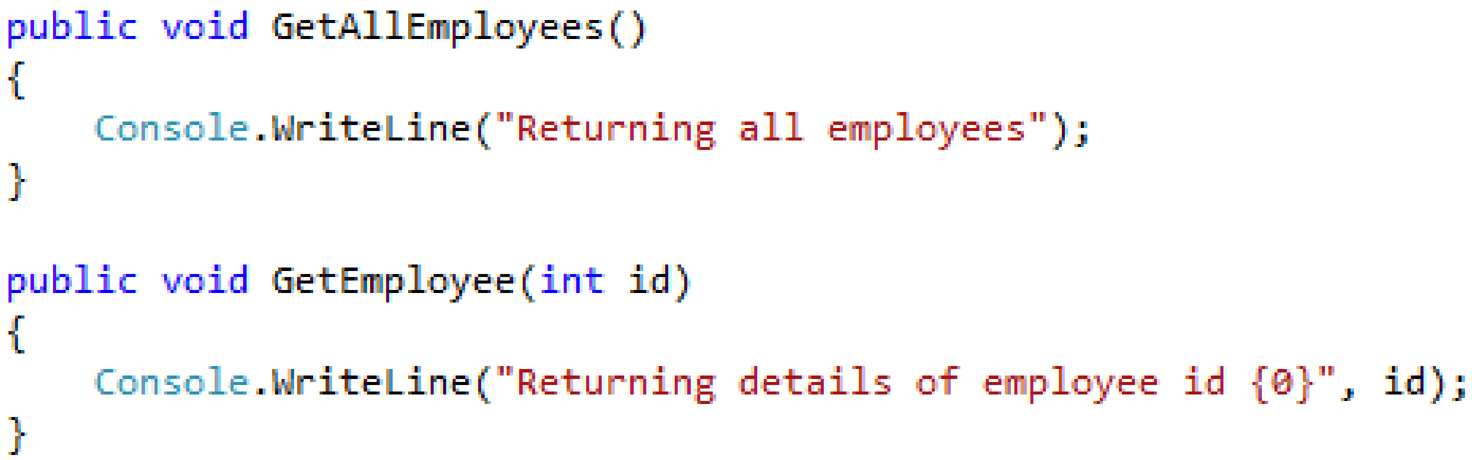


```
class Employee
{
    //fields can be used by all methods
    public int employeeId;
    public string name;

    public void GetAllEmployees()
    {
        //local variable, scope is current method only
        int employeeCount = 100;
        //write code to return all employees
    }

    public void GetEmployee()
    {
        //write code to return a specific employee
    }
}
```

# Demo

# Methods

- A method is a block of code
- Can perform a task when called
- Excepts parameters and return values.



```
public void GetAllEmployees()
{
    Console.WriteLine("Returning all employees");
}

public void GetEmployee(int id)
{
    Console.WriteLine("Returning details of employee id {0}", id);
}
```

# Methods and Modifiers

- Methods express behavior of a class
- Keywords change that behavior
  - Public
  - Private
  - Virtual
  - Static
- Keywords also controls arguments
- Parameter modifiers
  - None
  - Out
  - Ref
  - Params

# Bibliography, Important Links

- https://msdn.microsoft.com/en-us/library/67ef8sbd.aspx
- https://msdn.microsoft.com/en-us/library/ff926074.aspx

# Any Questions?

# Thank you!