Programming in C#. Fundamentals

AGENDA

- Lesson 1. Introduction to C# and .NET, Classes, Objects and Types
- Lesson 2. Classes and their main features
- Lesson 3. Control Flow
- Lesson 4. C# Best Practices: Collections and Generics
- Lesson 5. Object Oriented Programming
- Lesson 6. Threads, strings and regular expressions
- Lesson 7. Unit tests

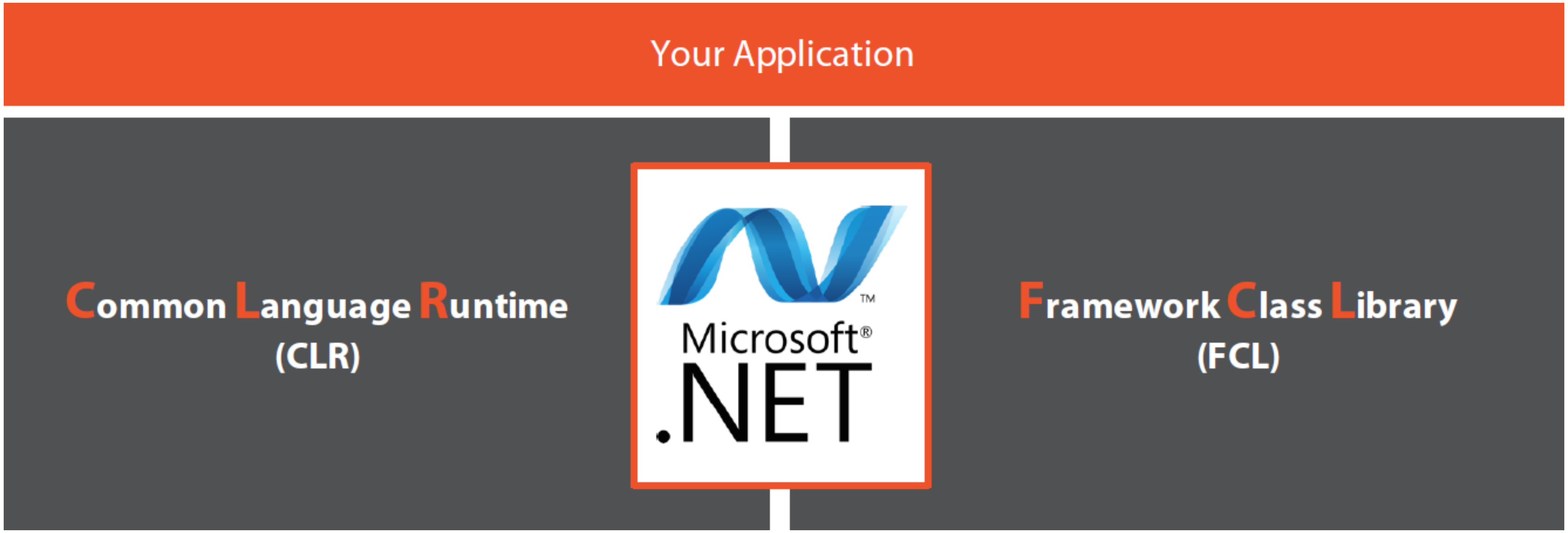
Lesson 1

Introduction to C# and .NET , Classes, Objects and Types

Introduction to C# and .NET

.NET Framework
The C# Language
Visual Studio

.NET is a software framework



Your Application

Common Language Runtime (CLR)

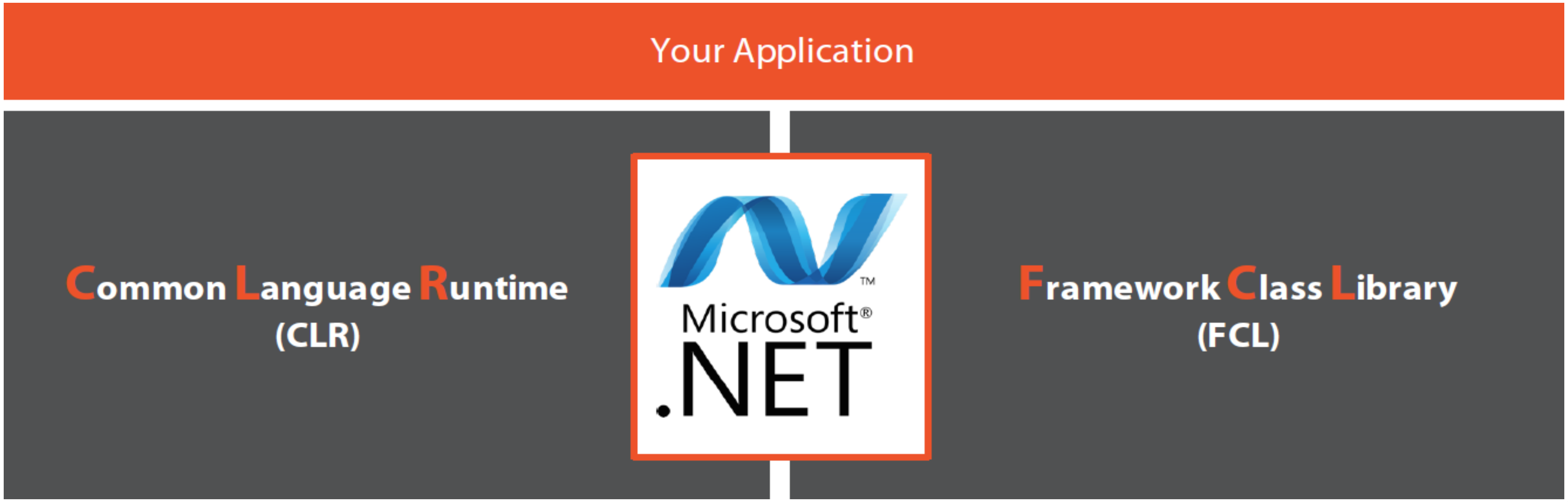
Microsoft® .NET™

Framework Class Library (FCL)

CLR

The CLR manages your application

- Memory management
- Operating system and hardware independence
- Language independence



Your Application

Common Language Runtime (CLR)

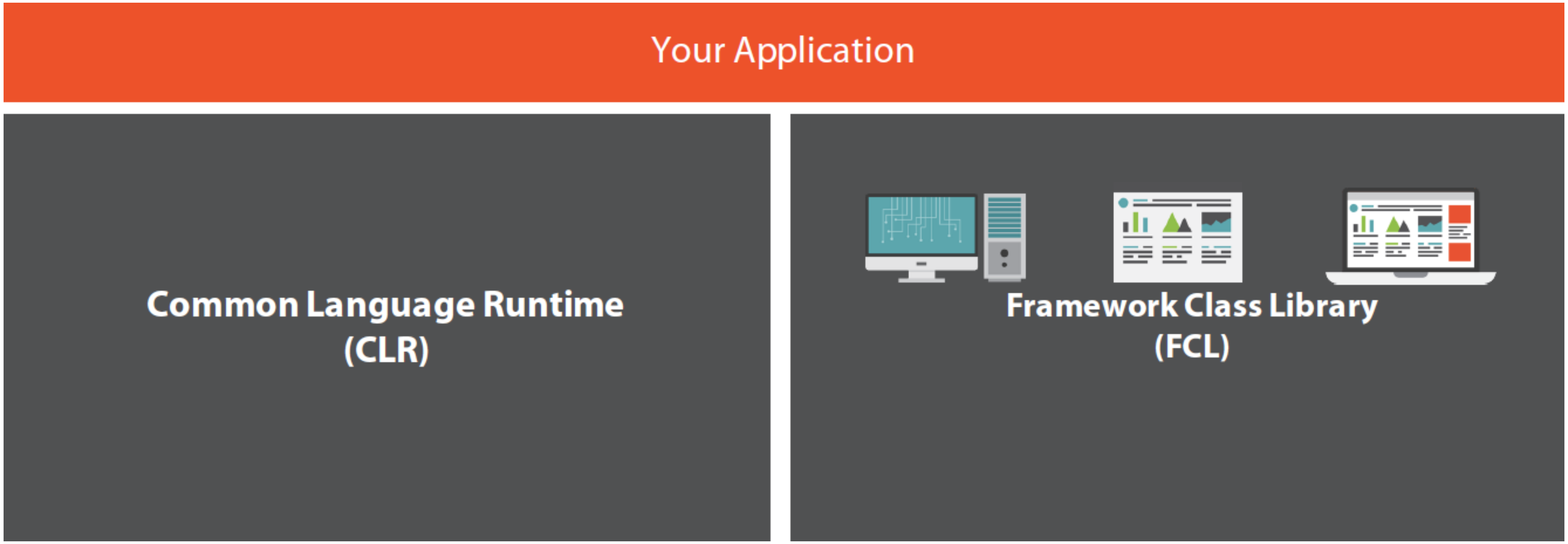
Microsoft® .NET™

Framework Class Library (FCL)

FCL

Framework class library

- A library of functionality to build applications



Your Application

Common Language Runtime (CLR)

Framework Class Library (FCL)

C#

One of many languages for .NET

- Syntax is similar to Java, C++, and JavaScript

C# is mixture between C++, Java and Delphi

- Fully object-oriented by design

Component-oriented programming model

  - Components, properties and events
  - Suitable for GUI and Web applications
  - XML based documentation

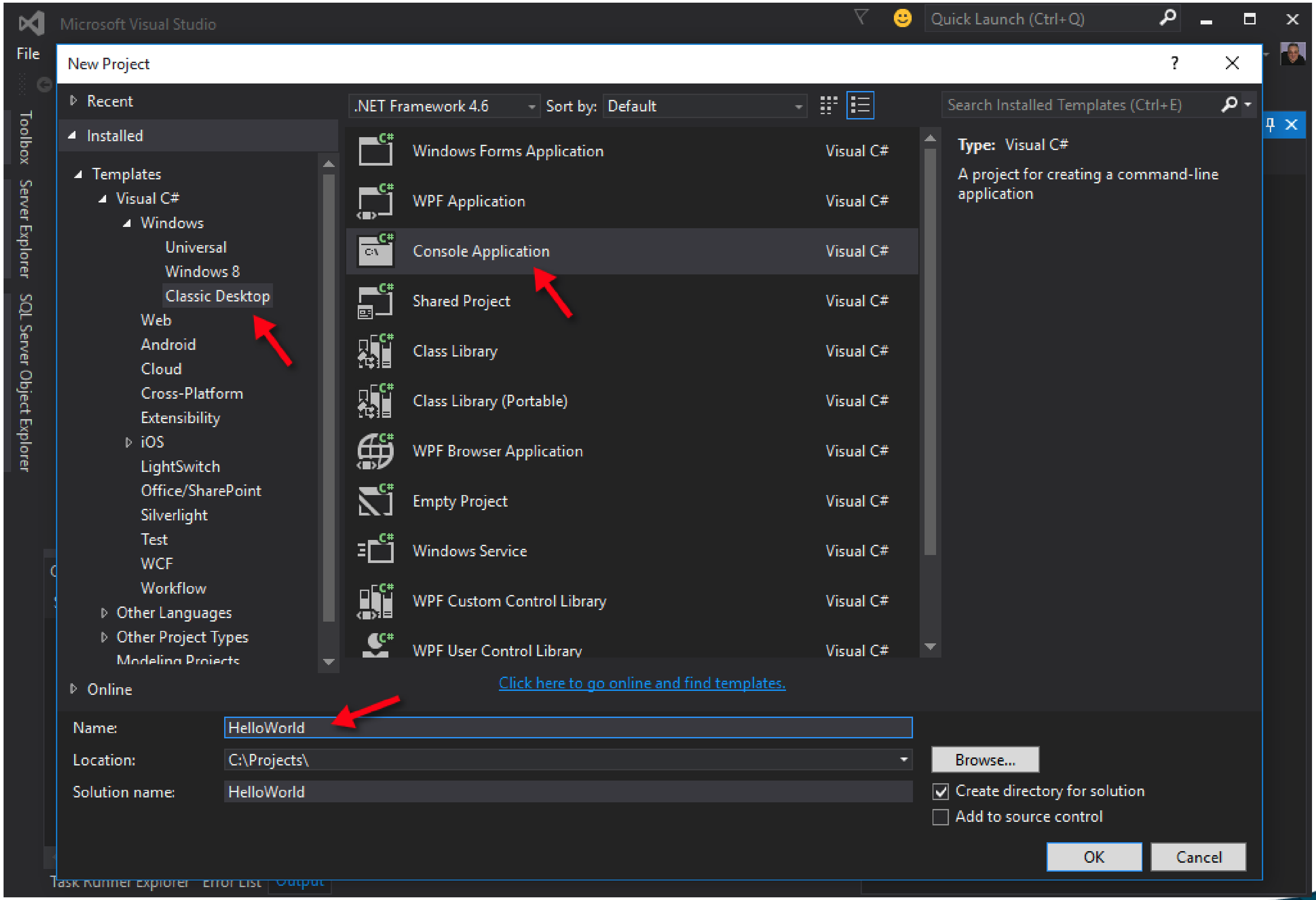
In C# all data types are objects

VISUAL STUDIO

Visual Studio is powerful Integrated Development Environment (IDE) for .NET Developers

- Create, edit, compile and run .NET applications
- Different languages – C#, C++, VB.NET, J#, …
- Flexible code editor
- Powerful debugger
- Integrated with SQL Server and IIS
- Strong support of Web services, WCF and WWF

Creating Hello World



New Project

Type: Visual C#

A project for creating a command-line application

Name: HelloWorld

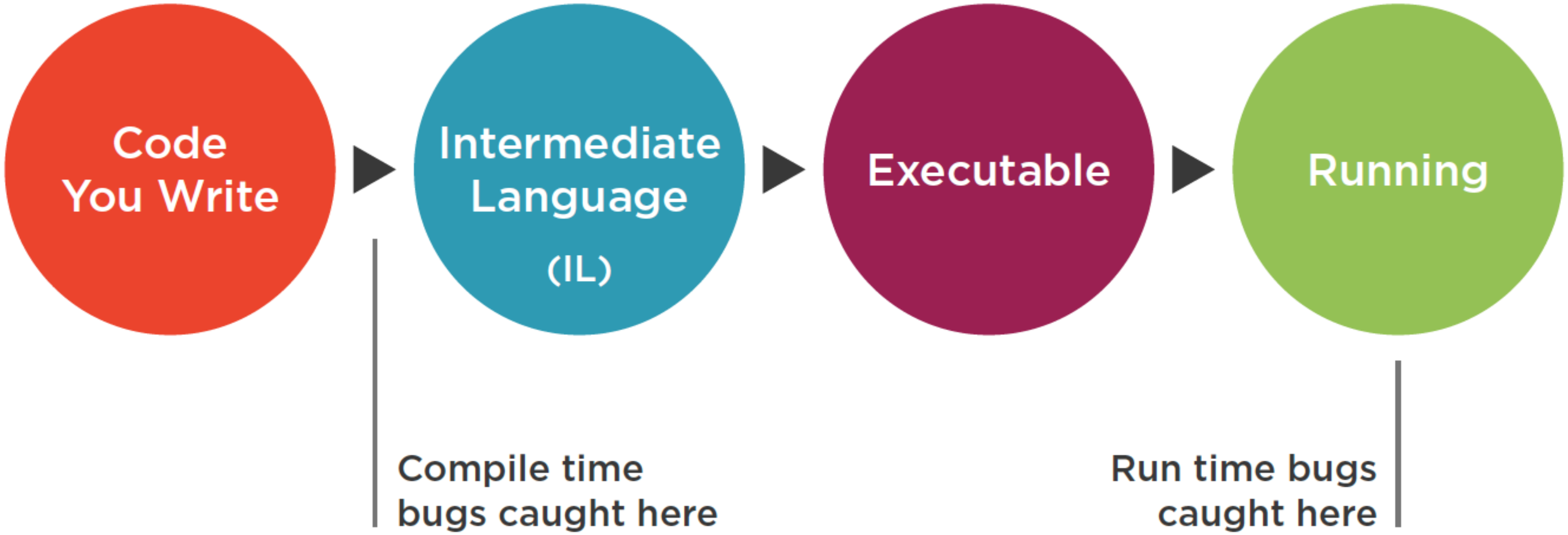
Location: C:\Projects\

Solution name: HelloWorld

☑ Create directory for solution

□ Add to source control

Word about compilation



Code You Write

Intermediate Language (IL)

Executable

Running

Compile time bugs caught here

Run time bugs caught here

Classes, Objects and Types

Classes and Objects

Keywords

Constructors

Classes and Variables

Access modifiers

Reference Types

Value Types

Struct and enum

C# Program Structure

Namespaces

- Contain types and other namespaces

Type declarations

- Classes, structs, interfaces, enums, and delegates

Members

- Constants, fields, methods, properties, indexers, events, operators, constructors, destructors

Keywords

as	explicit	null	switch
base	extern	object	this
bool	false	operator	throw
break	finally	out	true
byte	fixed	override	try
case	float	params	typeof
catch	for	private	uint
char	foreach	protected	ulong
checked	goto	public	unchecked
class	if	readonly	unsafe
const	implicit	ref	ushort
continue	in	return	using
decimal	int	sbyte	virtual
default	interface	sealed	volatile
delegate	internal	short	void
do	is	sizeof	while
double	lock	stackalloc	
else	long	static	
enum	namespace	string	

https://msdn.microsoft.com/en-us/library/x53a06bb(v=vs.71).aspx

Type System

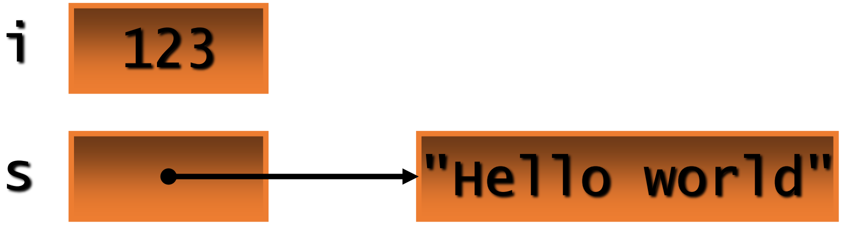
Value types

- Directly contain data
- Cannot be null

Reference types

- Contain references to objects
- May be null

```
int i = 123;
string s = "Hello world";
```



Type System

Value types

Primitives	`int i;`
Enums	`enum State { Off, On }`
Structs	`struct Point { int x, y; }`

Reference types

Classes	`class Foo: Bar, IFoo {...}`
Interfaces	`interface IFoo: IBar {...}`
Arrays	`string[] a = new string[10];`
Delegates	`delegate void Empty();`

Predefined Types

C# predefined types

Reference	object, string
Signed	sbyte, short, int, long
Unsigned	byte, ushort, uint, ulong
Character	char
Floating-point	float, double, decimal
Logical	bool

Predefined types are simply aliases for system-provided types

- For example, int == System.Int32

Variables

- A variable is a value holder
- It has a type
- It has an identifier
- It has a value that may change during the running of the program

Classes

Single inheritance

Multiple interface implementation

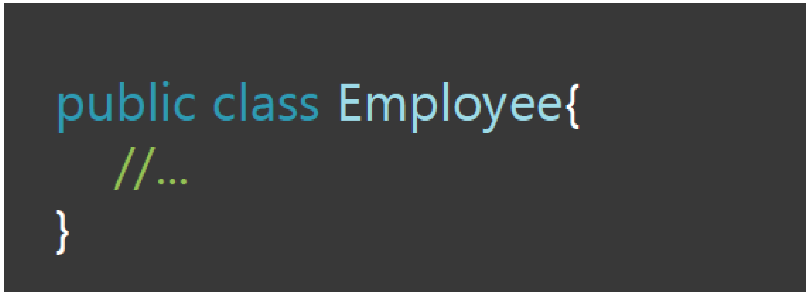
Class members

- Constants, fields, methods, properties, indexers, events, operators, constructors, destructors
- Static and instance members
- Nested types

Member access

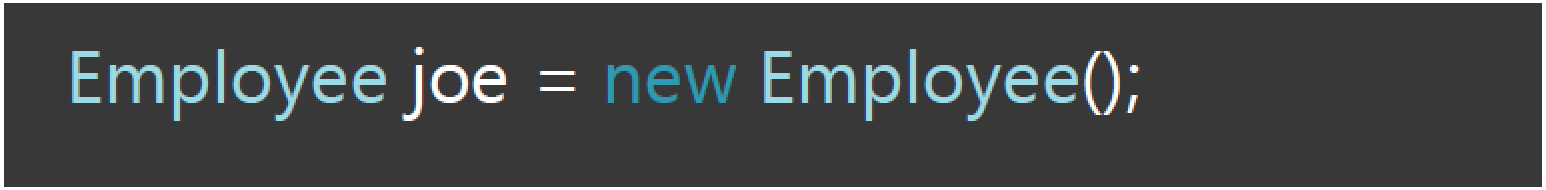
- public, protected, internal, private

Defining a Class



```
public class Employee{
    //...
}
```

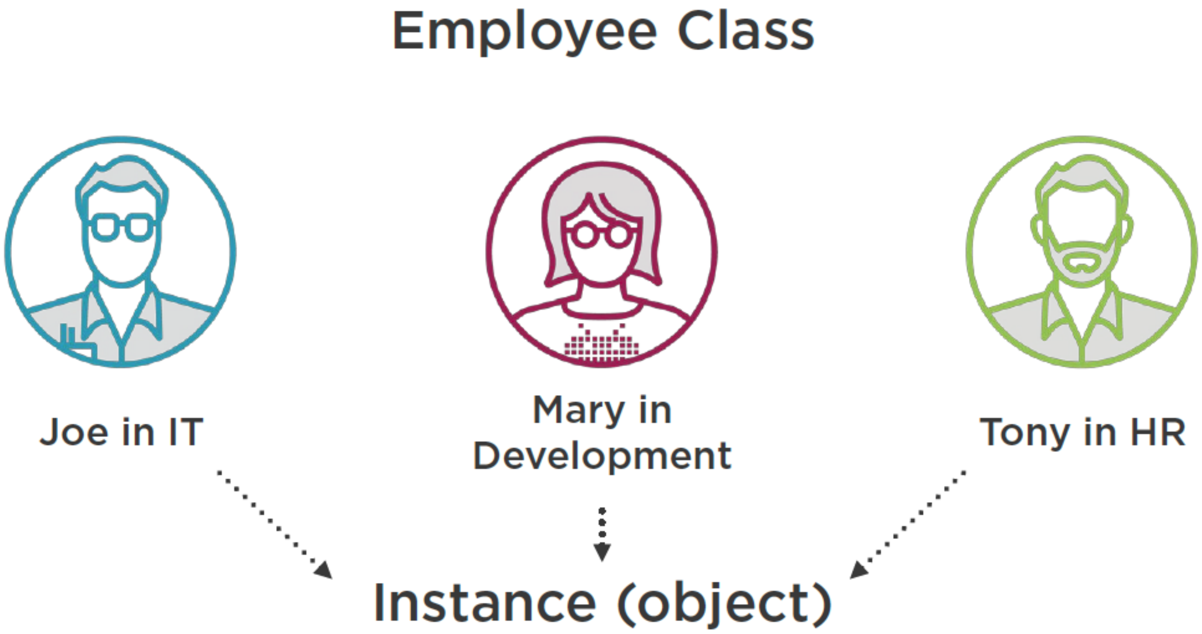
Instantiating an Object



```
Employee joe = new Employee();
```

An instance of a class is called an object

Classes and Objects



Employee Class

Joe in IT

Mary in Development

Tony in HR

Instance (object)

Structs

Like classes, except

- Stored in-line, not heap allocated
- Assignment copies data, not reference
- No inheritance

Ideal for light weight objects

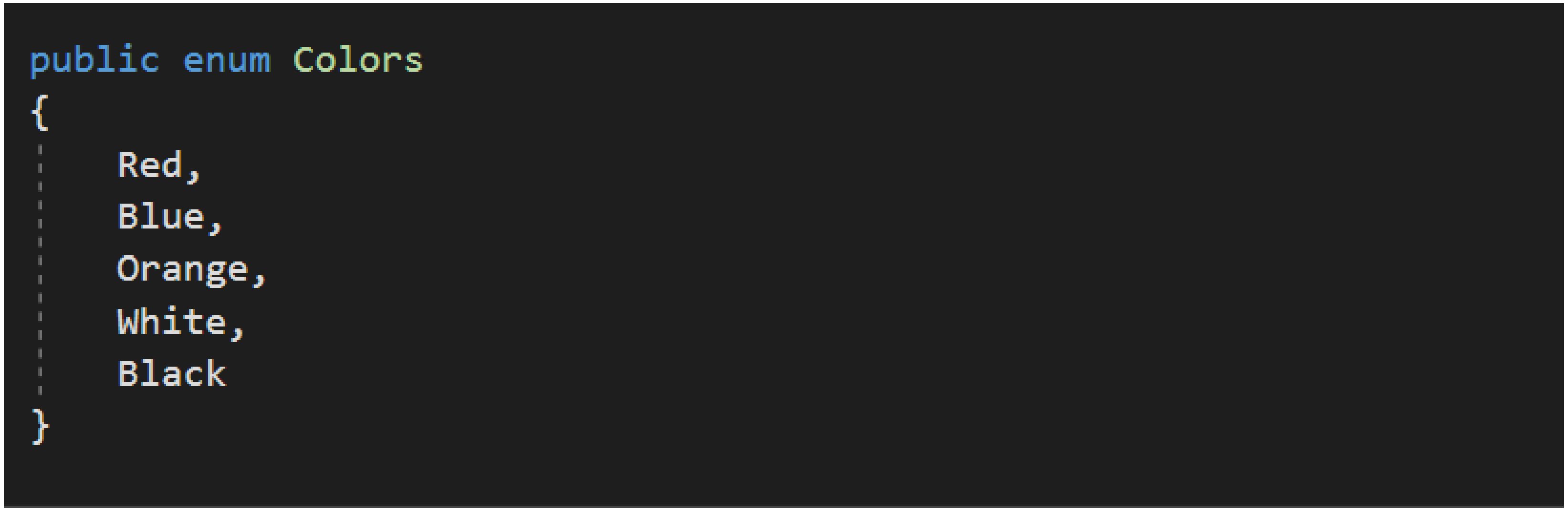
- Complex, point, rectangle, color
- int, float, double, etc., are all structs

Benefits

- No heap allocation, less GC pressure
- More efficient use of memory

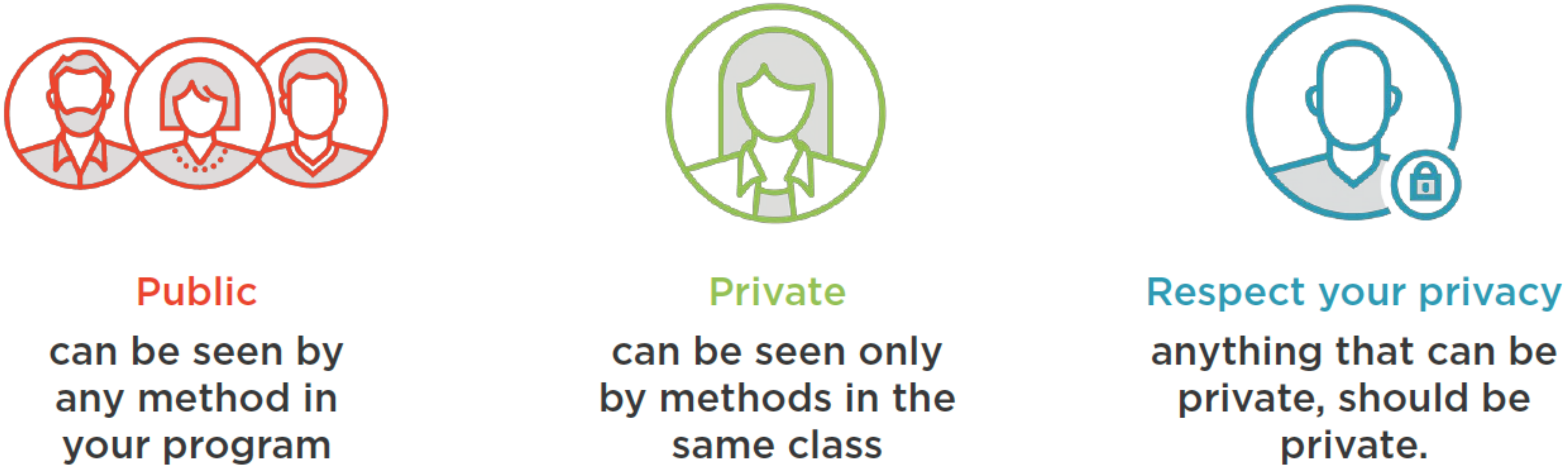
Enums

The enum keyword is used to declare an enumeration, a distinct type that consists of a set of named constants called the enumerator list.



```
public enum Colors
{
    Red,
    Blue,
    Orange,
    White,
    Black
}
```

A Word About Access Modifiers



Public

can be seen by any method in your program

Private

can be seen only by methods in the same class

Respect your privacy

anything that can be private, should be private.

Q & A

Practice Lesson 1

Home work