



# Programming Basics

Dieter Mourisse

# Praktische organisatie

# Lesschema

2 week 38				
ma 20 sep.	08:30 - 12:00	Programming basics	BST.L.3.101	Mourisse Dieter
di 21 sep.	08:30 - 12:00	Programming basics	BST.L.3.101	Mourisse Dieter
3 week 39				
ma 27 sep.	08:30 - 12:00	Programming basics	BST.L.3.101	Mourisse Dieter
di 28 sep.	08:30 - 12:00	Programming basics	BST.L.3.101	Mourisse Dieter
4 week 40				
ma 4 okt.	08:30 - 12:00	Programming basics	BST.L.3.101	Mourisse Dieter
di 5 okt.	08:30 - 12:00	Programming basics	BST.L.3.101	Mourisse Dieter
5 week 41				
ma 11 okt.	08:30 - 12:00	Programming basics	BST.L.3.101	Mourisse Dieter
di 12 okt.	08:30 - 12:00	Programming basics	BST.L.3.101	Mourisse Dieter
6 week 42				
ma 18 okt.	08:30 - 12:00	Programming basics	BST.L.3.101	Mourisse Dieter
di 19 okt.	08:30 - 12:00	Programming basics	BST.L.3.101	Mourisse Dieter
7 week 43				
ma 25 okt.	08:30 - 12:00	Programming basics	BST.L.3.101	Mourisse Dieter
di 26 okt.	08:30 - 12:00	Programming basics	BST.L.3.101	Mourisse Dieter

# Inhoud

---

In deze module verkrijg je een grondig inzicht in de werking van het .Net Framework en leer je op een gestructureerde manier programma's te ontwikkelen. Je verwerft inzicht in de principes van object georiënteerd programmeren.

- Variables
- Methods
- Programming structures
  - Selection
  - Iteration
- Enums
- Arrays & Collections
- Error handling
- Classes
  - Fields
  - Properties
  - Methods
  - Accessors
- Debugging

[https://leho-howest.instructure.com/courses/13062/external\\_tools/15833](https://leho-howest.instructure.com/courses/13062/external_tools/15833)

# Evaluatie

---

*Evaluatie(s) voor de eerste examenkans*

Moment	Vorm	%	Opmerking
examenperiode 1 (1e sem) binnen examenrooster	examen: andere vorm of combinatie van vormen	60,00	
examenperiode 1 buiten examenrooster	permanente evaluatie: andere vorm of combinatie van vormen (Permanente evaluatie)	40,00	Een gemiddelde score van 8/20 op de permanente evaluatie is vereist om te kunnen slagen voor deze module in EK1.



.NET

# Hoofdstuk 1: Inleiding

# Enkele begrippen

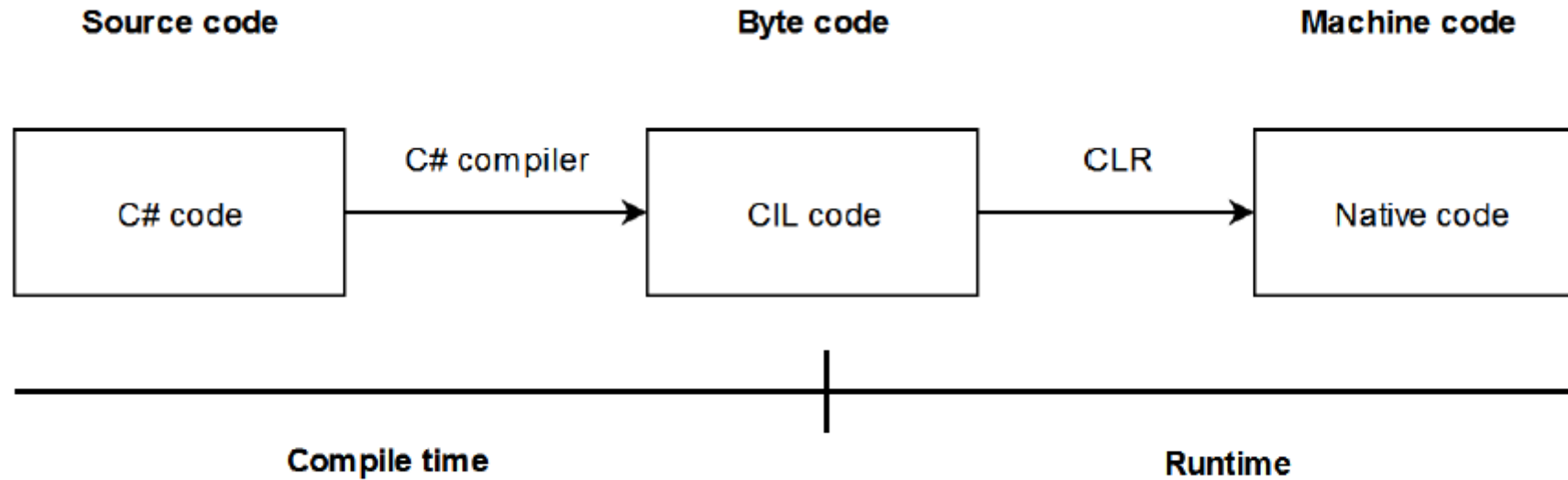
---

Term	Betekenis
C Sharp (C#)	De programmeertaal die we gebruiken
.NET	Een verzamelnaam voor verschillende frameworks
.NET Framework	Originele framework, geleverd samen met windows
.NET Core	Een cross-platform versie van .NET Framework
Xamarin (Mono.Net)	Framework voor mobiele apps

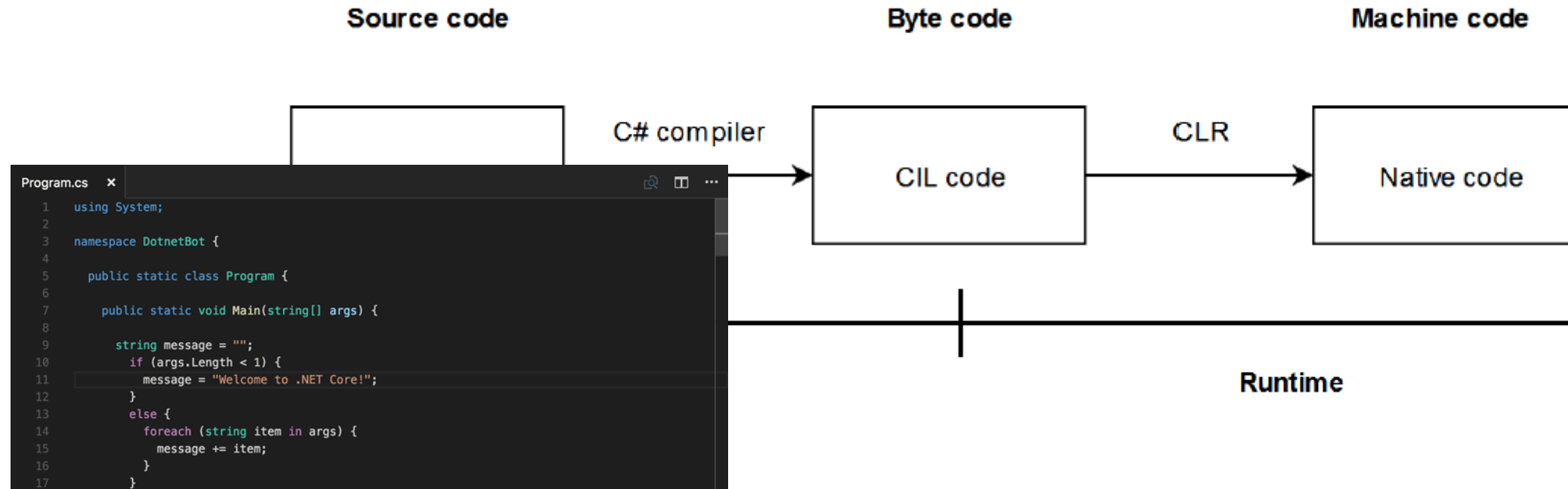


<b>.NET FRAMEWORK</b>	<b>.NET CORE</b>	<b>XAMARIN</b>
Platform for .NET applications on Windows	Cross-platform and open source framework optimized for modern app needs and developer workflows	Cross-platform and open source Mono-based runtime for iOS, OS X, and Android devices
Distributed with Windows	Distributed with app	Distributed with app

# Van code tot uitvoering



# Van code tot uitvoering



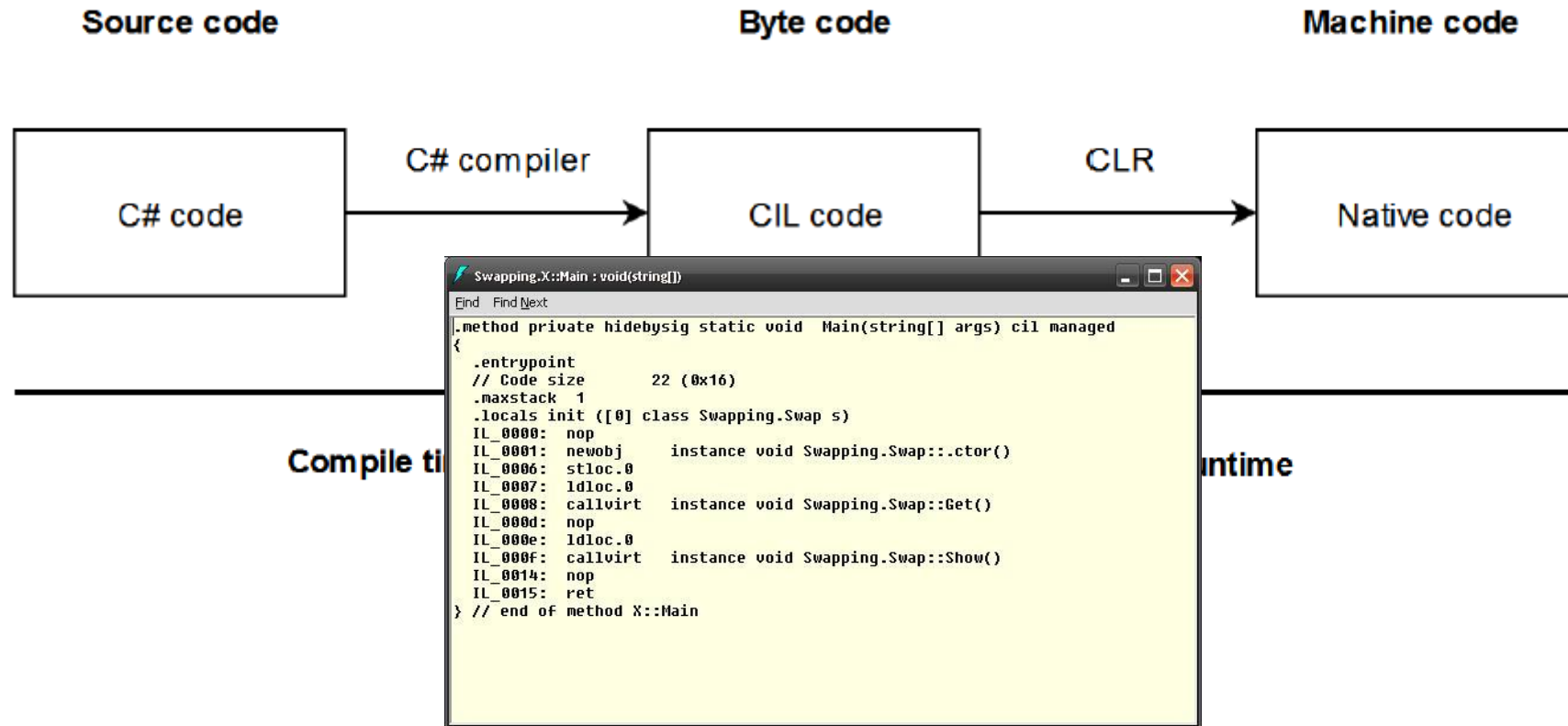
# Van code tot uitvoering

Program.cs x



```
1  using System;
2
3  namespace DotnetBot {
4
5      public static class Program {
6
7          public static void Main(string[] args) {
8
9              string message = "";
10             if (args.Length < 1) {
11                 message = "Welcome to .NET Core!";
12             }
13             else {
14                 foreach (string item in args) {
15                     message += item;
16                 }
17             }
18         }
19     }
20 }
```

# Van code tot uitvoering



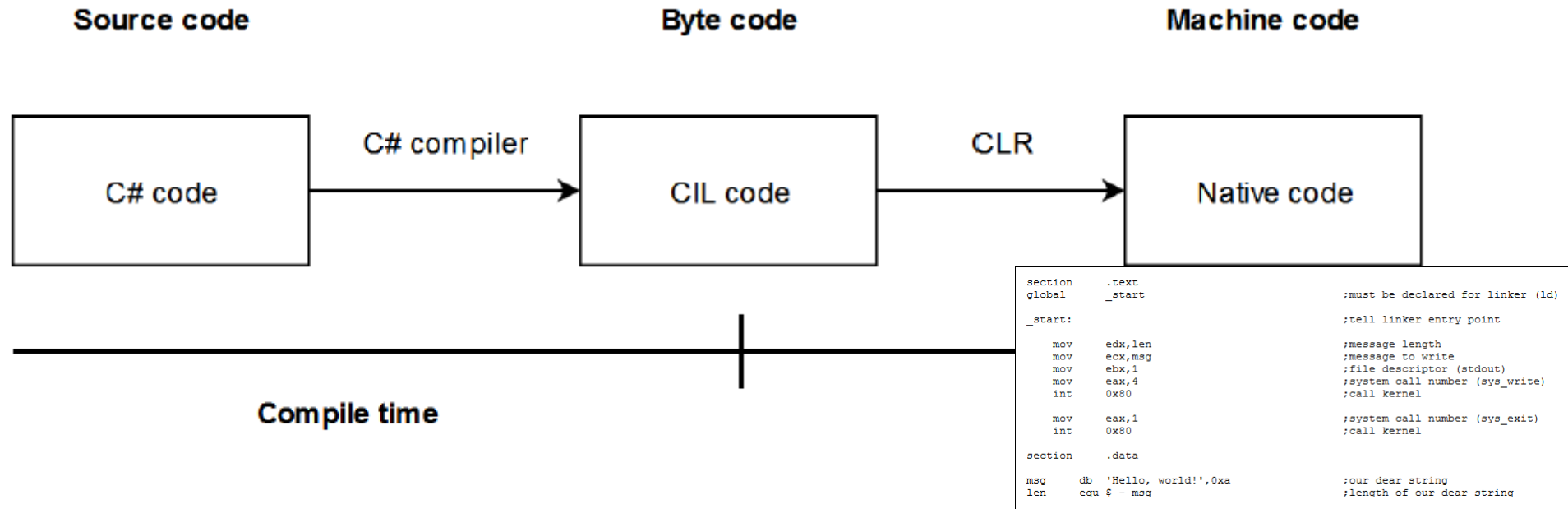
Van code



```
Swapping.X::Main : void(string[])
Find Find Next

.method private hidebysig static void Main(string[] args) cil managed
{
    .entrypoint
    // Code size          22 (0x16)
    .maxstack 1
    .locals init ([0] class Swapping.Swap s)
    IL_0000:  nop
    IL_0001:  newobj      instance void Swapping.Swap::.ctor()
    IL_0006:  stloc.0
    IL_0007:  ldloc.0
    IL_0008:  callvirt    instance void Swapping.Swap::Get()
    IL_000d:  nop
    IL_000e:  ldloc.0
    IL_000f:  callvirt    instance void Swapping.Swap::Show()
    IL_0014:  nop
    IL_0015:  ret
} // end of method X::Main
```

# Van code tot uitvoering



# Van code tot uitvoering

```
section      .text
global      _start                ;must be declared for linker (ld)

_start:                                           ;tell linker entry point

    mov     edx,len                ;message length
    mov     ecx,msg                ;message to write
    mov     ebx,1                  ;file descriptor (stdout)
    mov     eax,4                  ;system call number (sys_write)
    int     0x80                  ;call kernel

    mov     eax,1                  ;system call number (sys_exit)
    int     0x80                  ;call kernel

section      .data

msg          db  'Hello, world!',0xa            ;our dear string
len          equ $ - msg                       ;length of our dear string
```





# Visual Studio