C# fundamentals

Agenda

What is C# ?

Compilation and execution

Keywords

Create a console application

Statements

Blocks

Variables

Implicitly typed variables

Expressions

IntelliSense

Classes and members

Write to the console

What is C# ?

C# is an elegant and type-safe object-oriented language. C# enables developers to build many types of secure and robust applications that run in the .NET ecosystem

C# Programming Guide

C#

One of the most popular programming languages

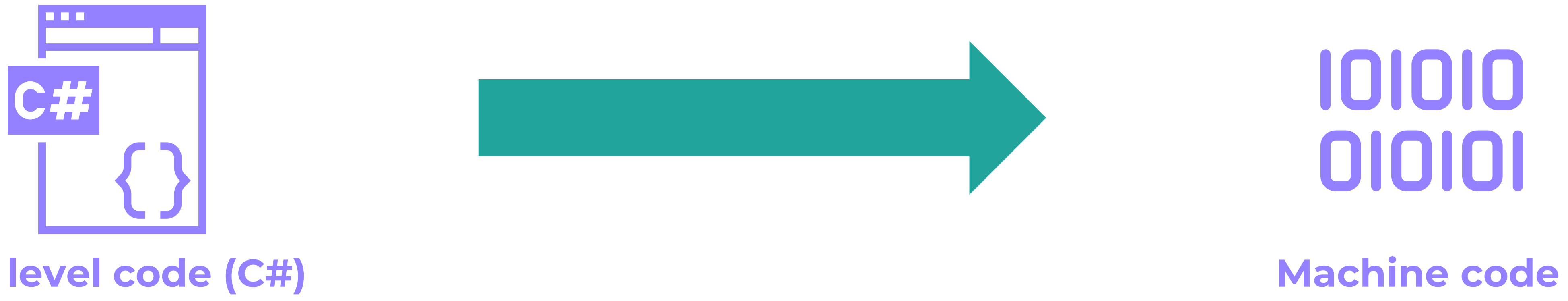
Similar to Java or C++ in its syntax

What is C# ?

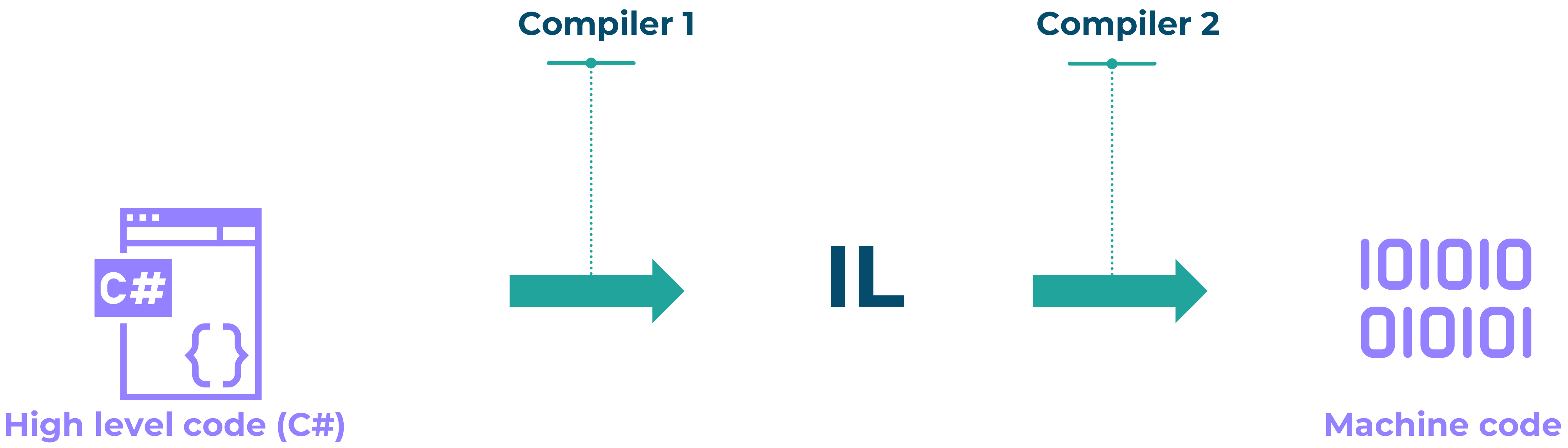
- Object oriented
- Type safe
- Standardized
- Advanced features

Compilation and execution

Compilation process



Compilation process in .NET



IL

Platform independant

Independant from the programming language

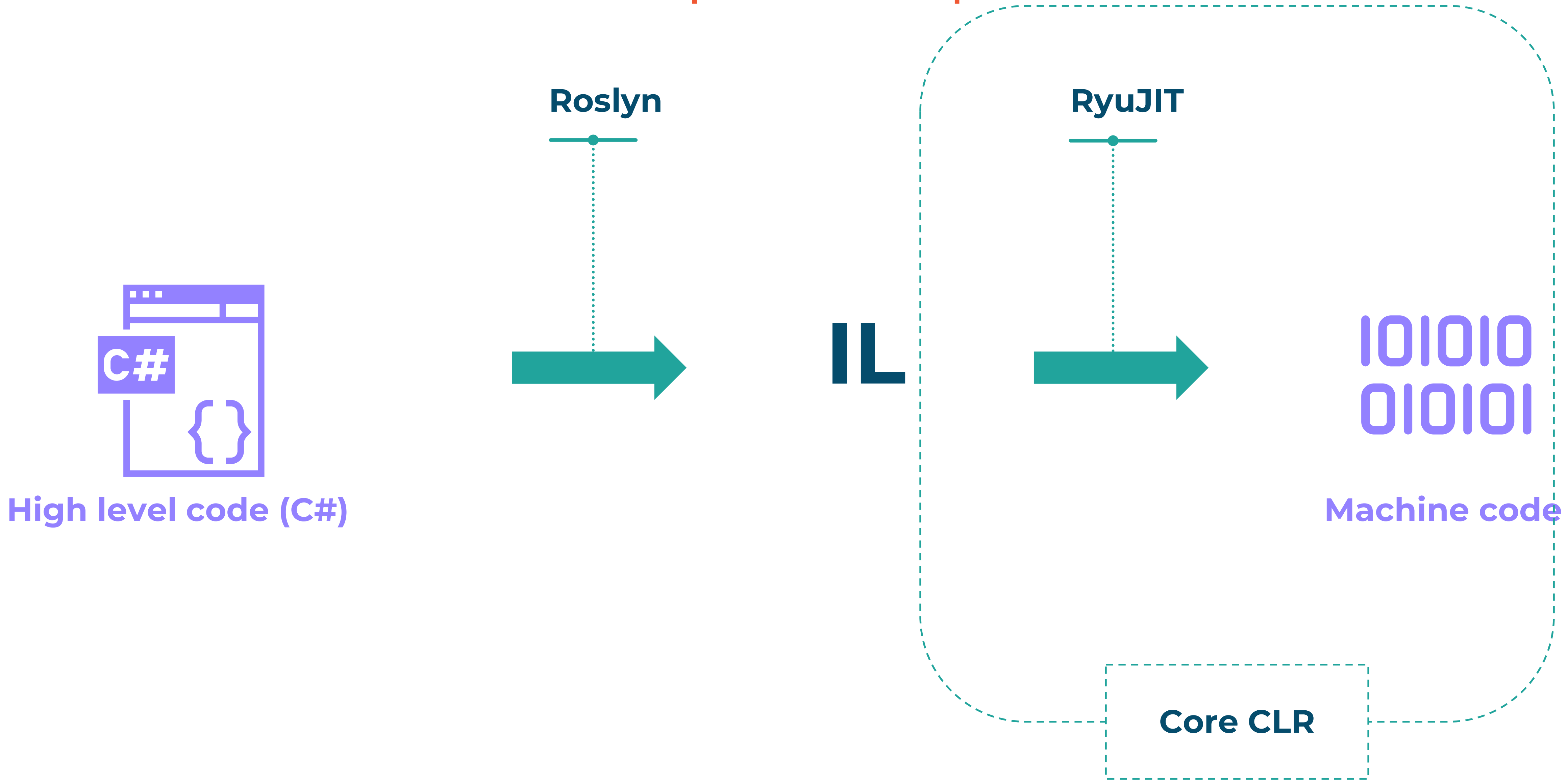
Allows cross language integration support and cross platform support

CLR

Manages :

- Objects and memory
- Assembly loading
- Exceptions (errors)
- Security
- Object lifetime (garbage collection)
- Type safety
- …

C# compilation process



Demo

Compile and run C# code

Compile and runtime errors

Create a console application

Demo

Create a console application

Keywords

Keywords

Keywords are reserved words

They have a special meaning to the compiler

Keywords

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

Demo

Use some C# keywords

Statements

Statements

The actions that a program takes are expressed in statements

A block of code can be divided into a set of statements

Statements

A statement can be :

- A single line of code ending with a semicolon (variable declaration, method call)
- Multiple lines of code in a block inside { } brackets (loop)

Demo

Single line statement

Multiple lines statement

Blocks

Blocks

A block is a boundary which is defined with curly brackets

A block can delimit a namespace, a class, a method or a statement

Demo

C# blocks

Variables

Variables

A variable is a holder that contains data

It is stored physically in memory

The data stored in memory is the value

Built-in types

Integral and floating-point numeric types (`int`, `float`)

Decimals (`decimal`)

Booleans (`bool`)

Characters (`char`)

String of characters (`string`)

Variable declaration

```
bool myBoolean = true;
int myInteger = 7;
string myString = "Hello!";
```

Demo

Create variables

Implicitly typed variables

Implicitly typed variables

The var keyword replaces the type

The type of the variable is inferred from its value

Cannot be initialized to null

Implicitly typed variable declaration

```
var myBoolean = true;
var myInteger = 7;
var myString = "Hello!";
```

Demo

Create implicitly typed variables

Operators

Operators

Perform basic operations on built-in types

Operators

Operators covered in this course:

- Arithmetic
- Comparison
- Logical
- Equality

Comparison operators

Greater than

Lower than

Greater than or equal

Lower than or equal

Logical operators

Negation operator !

AND &

Exclusive OR ^

Conditional AND

Conditional OR

Equality operators

Equality ==

Inequality !=

String concatenation

+ operator concatenates `string` variables

+= operator

Concatenates `string`

Addition assignment operator on integers(`int`,...) or floating-point types (`float`,...)

Demo

Use covered operators

Expressions

Expressions

An expression is a sequence of operators and operands

There are different types of expressions

Expressions can be combined to make larger expressions

Expressions

Boolean expression

Variable declaration

Literal, variable assignment

Namespace declaration

`new` operator expression

Initializer

Lambda expression

Boolean expression

```
1 > 0
```

Boolean expression

```
if (1 > 0)
{
    // code
}
```

Boolean expression

```
string text = "I love C#!";
if (1 > 0 && text.Length == 10)
{
    // code
}
```

Literal

```
string myString = "I love C#!";
```

Literal

```
bool boolean = true;
```

Variable declaration

```
string myString;
```

Variable assignment

```
string text = "I love C#!";
```

Variable assignment

```
bool boolean = 1 > 0;
```

new expression

```
var date = new DateTime(2000, 1, 1, 0, 0, 0);
```

Object initializer

```
var movie = new Movie
{
    Id = 1,
    Title = "Movie title",
    Overview = "Movie overview"
};
```

Demo

Use IntelliSense on VS Code

Demo

Declare a class with its members

Instantiate the class

Summary

C# is a modern, object oriented and standardized programming language

C# code is compiled into an Intermediate Language (platform agnostic) then into machine code (platform specific)

C# provides reserved keywords that constitute the words of the language

C# code is structured into blocks of code

A block of code can contain statements

A Statement is a combination of expressions

Expressions are a combination of operators and operands (variables)

Variables contain the data used in your program

Summary

Each variable is physically stored in memory

A variable has a type

A variable can be typed explicitly or implicitly

C# provides some built-in types

Variables can be manipulated through operators (arithmetic, comparison, logical …)

You can create your own types by creating classes

A class can have fields, properties and methods (members)