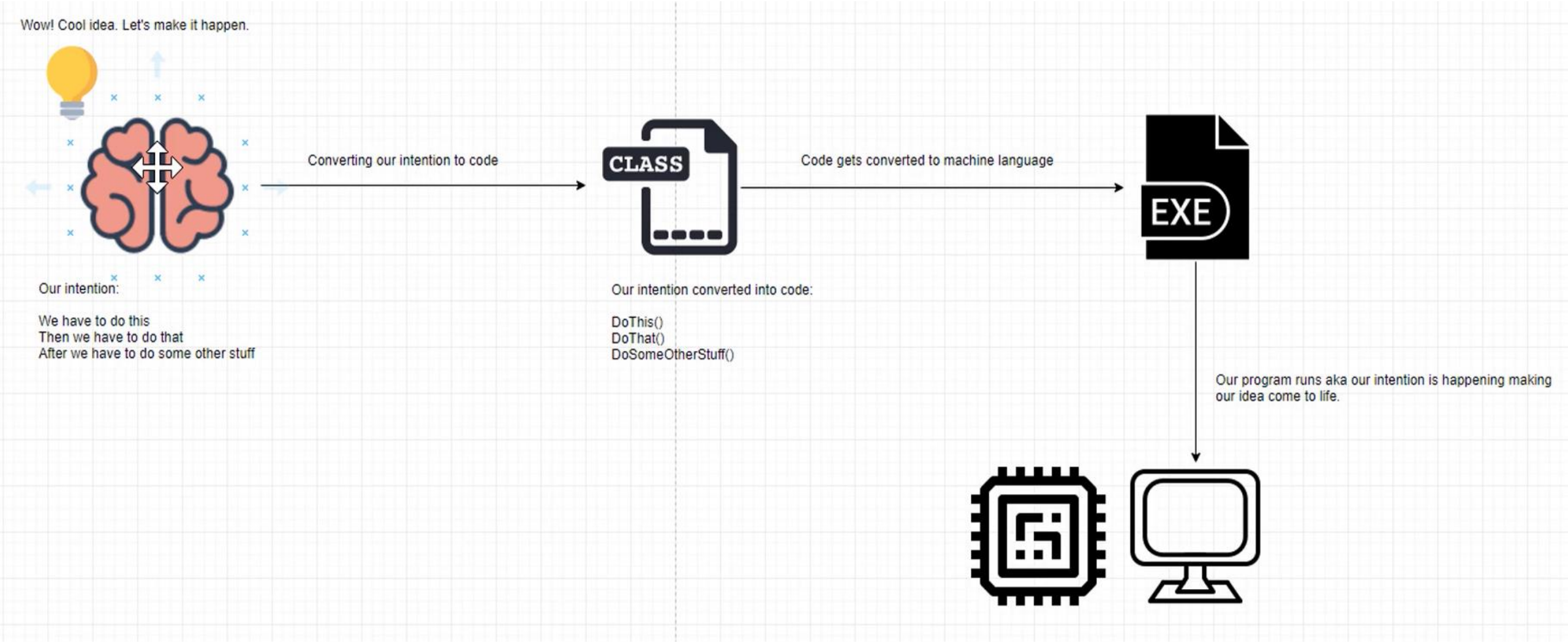
Object Oriented Programming Fundamental

Ridi Ferdiana | ridi@ugm.ac.id
Version 6.0.0

What is Program?

Avetis Ghukasyan, 2020



Computer Codes (Low Level)

Machine code	Assembly code	Description
001 1 000010	LOAD #2	Load the value 2 into the Accumulator
010 0 001101	STORE 13	Store the value of the Accumulator in memory location 13
001 1 000101	LOAD #5	Load the value 5 into the Accumulator
010 0 001110	STORE 14	Store the value of the Accumulator in memory location 14
001 0 001101	LOAD 13	Load the value of memory location 13 into the Accumulator
011 0 001110	ADD 14	Add the value of memory location 14 to the Accumulator
010 0 001111	STORE 15	Store the value of the Accumulator in memory location 15
111 0 000000	HALT	Stop execution

Intermediate Language

IIF Örneği

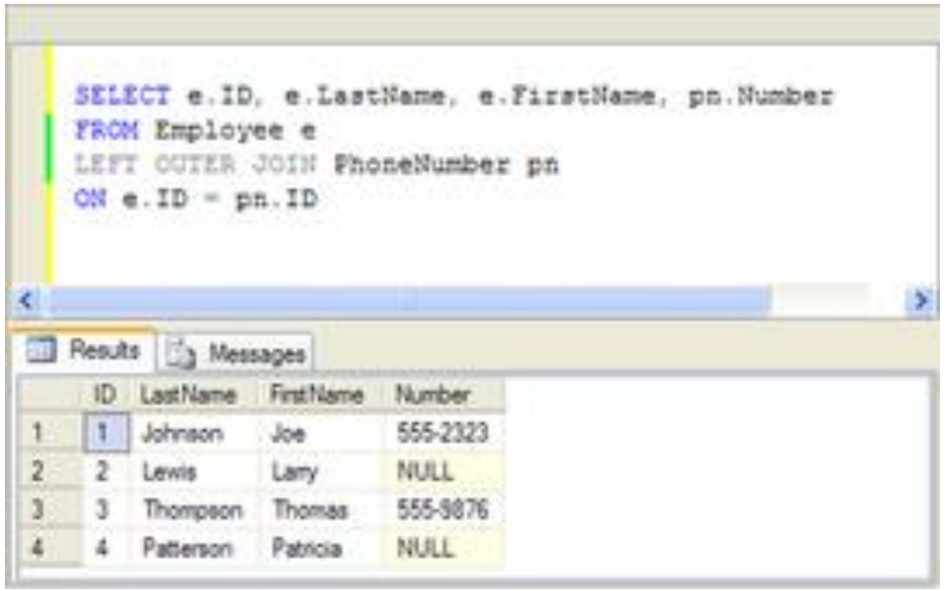
```
.method private instance void Button1_Click
(object sender, class [mscorlib]System.EventArgs e)
cil managed
{
   .maxstack 3
   .locals init (
      [0] object Mesaj,
      [1] bool Oldu)
   L_0000: nop
   L_0001: ldc.i4.1
   L_0002: stloc.1
   L_0003: ldloc.1
   L_0004: ldstr "Oldu"
   L_0009: ldstr "Olmad\u0131"
   L_000e: call object
[Microsoft.VisualBasic]
Microsoft.VisualBasic.Interaction:
:Iif(bool, object, object)
   L_0013: call object [mscorlib]
System.Runtime.CompilerServices.
RuntimeHelpers::GetObjectValue(object)
   L_0018: stloc.0
   L_0019: ldloc.0
   L_001a: call object [mscorlib]
System.Runtime.CompilerServices.RuntimeHelpers:
:GetObjectValue(object)
   L_001f: ldc.i4.0
   L_0020: ldnull
   L_0021: call valuetype [Microsoft.VisualBasic]
Microsoft.VisualBasic.MsgBoxResult
[Microsoft.VisualBasic]
Microsoft.VisualBasic.Interaction::
MsgBox(object, valuetype [Microsoft.VisualBasic]
Microsoft.VisualBasic.MsgBoxStyle, object)
   L_0026: pop
   L_0027: nop
   L_0028: ret
}
```

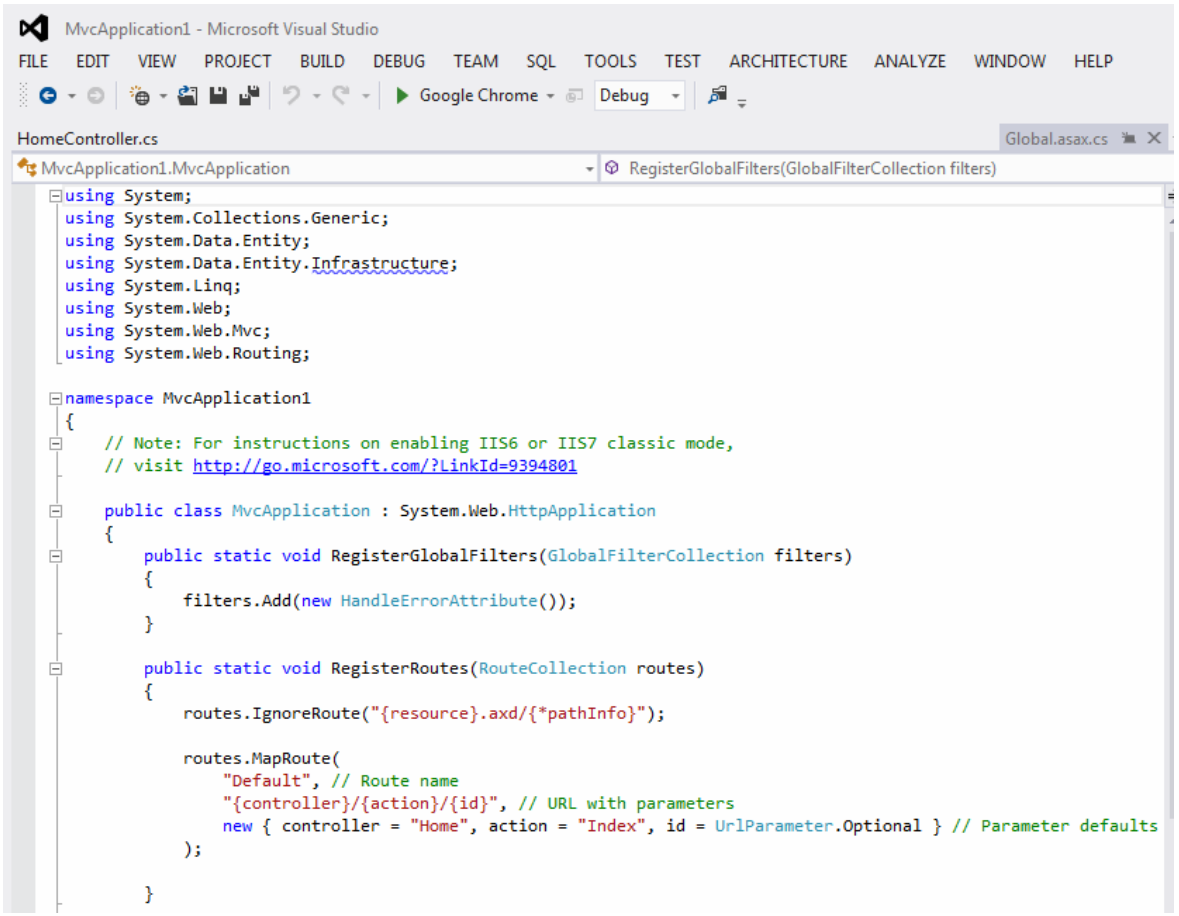
IF Örneği

```
.method private instance void Button1_Click
(object sender, class [mscorlib]System.EventArgs e)
cil managed
{
   .maxstack 3
   .locals init (
      [0] string Mesaj,
      [1] bool Oldu)
   L_0000: nop
   L_0001: ldc.i4.1
   L_0002: stloc.1
   L_0003: ldloc.1
   L_0004: brtrue.s L_000d
   L_0006: ldstr "Olmad\u0131"
   L_000b: br.s L_0012
   L_000d: ldstr "Oldu"
   L_0012: stloc.0
   L_0013: ldloc.0
   L_0014: ldc.i4.0
   L_0015: ldnull
   L_0016: call valuetype [Microsoft.VisualBasic]
Microsoft.VisualBasic.MsgBoxResult
[Microsoft.VisualBasic]
Microsoft.VisualBasic.Interaction::
MsgBox(object, valuetype [Microsoft.VisualBasic]
Microsoft.VisualBasic.MsgBoxStyle, object)
   L_001b: pop
   L_001c: nop
   L_001d: ret
}
```

High Level Language

SQL, C#, HTML, PHP, XML, JSON, etc.





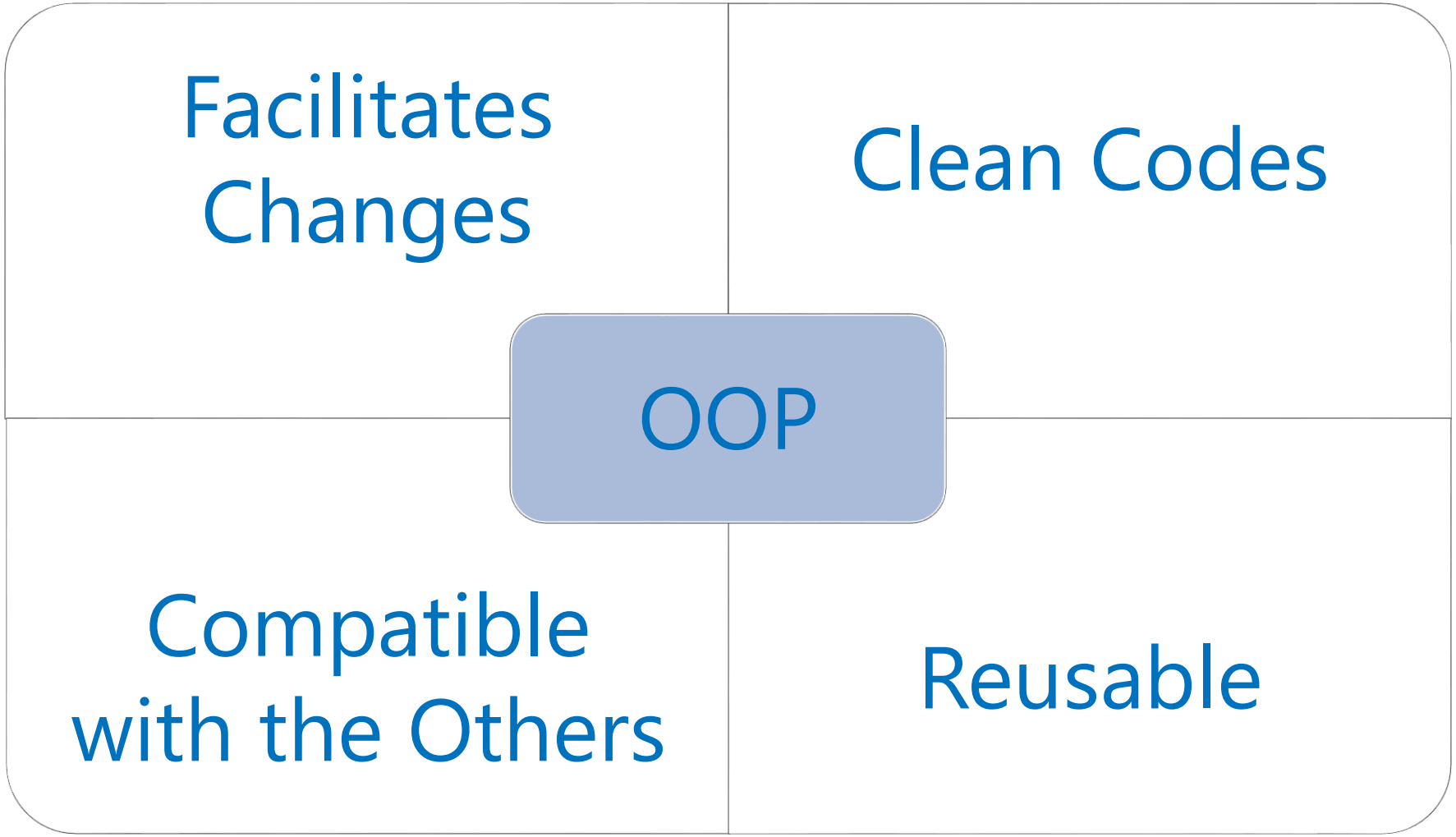
Low Codes Development

- Power Platforms (Microsoft)
- Honey Codes (Amazon)
- Lightning Codes (Salesforce)
- Creatio
- GeneXus
- Creator (Zoho)
- Appian

What is OOP

- An approach to software development in which the structure of the software is based on objects interacting with each other

OOP was developed to improve the **flexibility** of the software



Procedural Programming with OO Programming Language

Procedural Programming

- Organize the program in a top down approach

Procedural Benefits

Natural development

Simplicity of Concept

Might Be Faster to Execute

Popular

Can be applied with OO Language

Procedural Problems

Harder to maintain	Hard to Alter (Domino)	Modification from Scratch
Single Fighter Development	Translating Complex Business Problems	Limited Integration with Others System

Converting to OO Programming Language

OO Benefits

A more intuitive transition

The ability to maintain and implement changes

The ability to more effectively create software systems using a team process

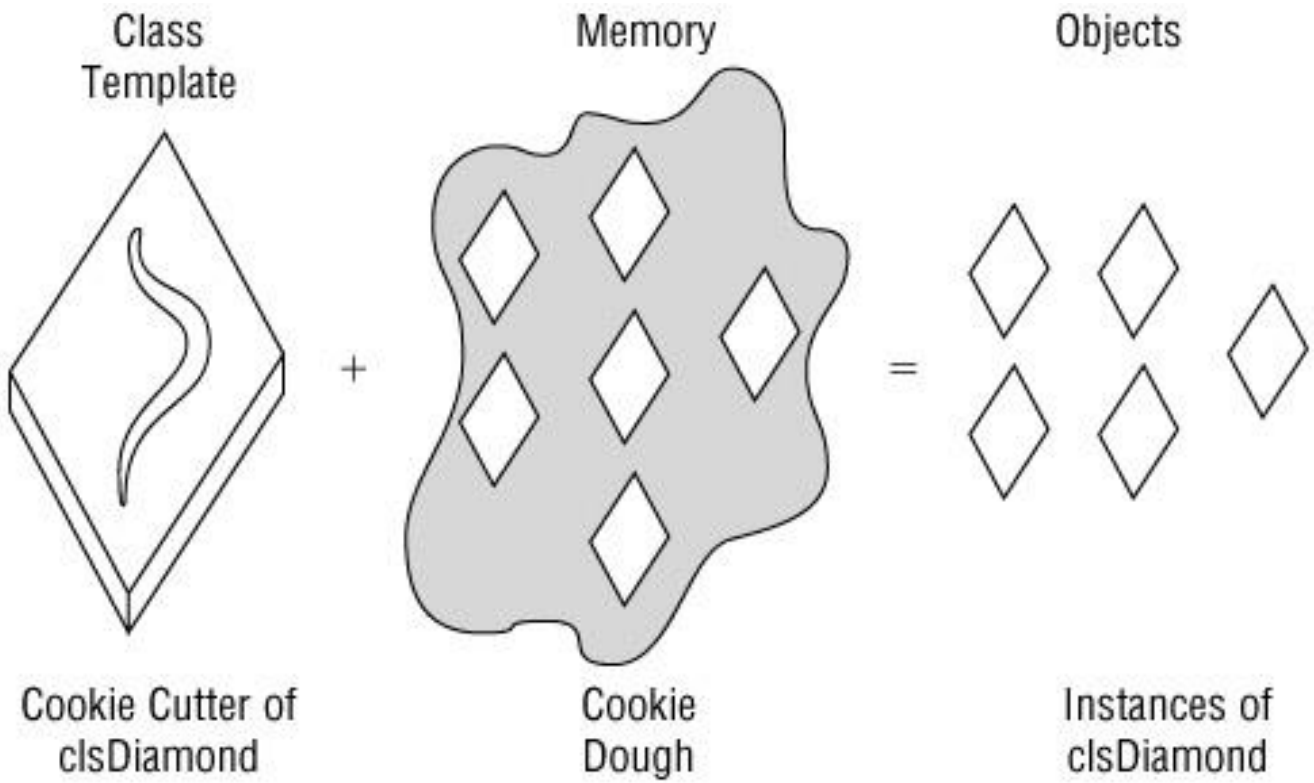
The ability to reuse code components

Better integration with loosely coupled distributed computing systems.

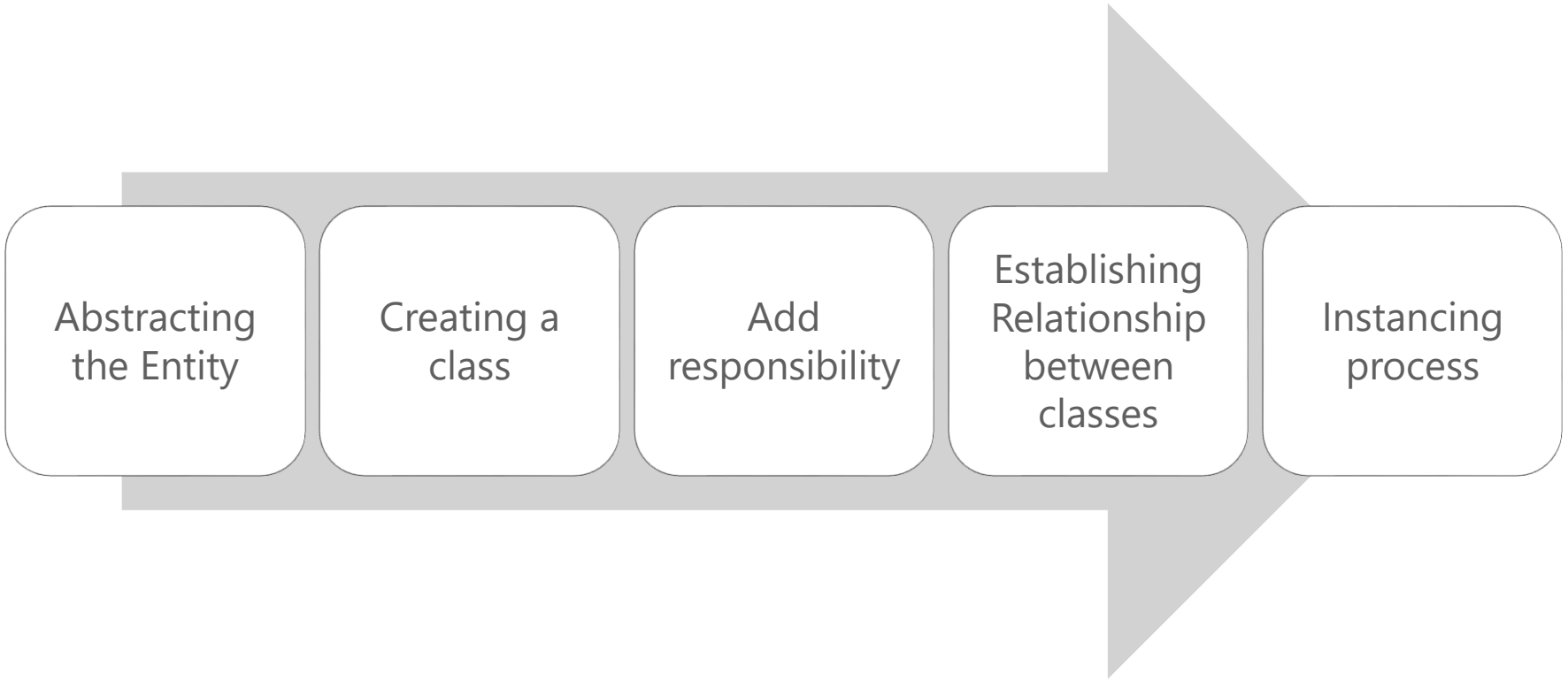
Class

Object (Entity) and its properties

Instance


Class
Template
Memory
Objects
+
=
Cookie Cutter of
clsDiamond
Cookie
Dough
Instances of
clsDiamond

OO in Pragmatic Way



.NET Core SDK – Object Oriented Features

Key points

- OO is a paradigm in software development to "Think in Object"
- OO focuses on object and its relationship.
- OO concepts are Entity, Class, Object
- Instance is a process from understanding the entity, designing the class, and creating the object.
- Three steps to implement OO paradigm which are entity identification, constructing a class, understanding its responsibilities, establishing relationships, and use the object