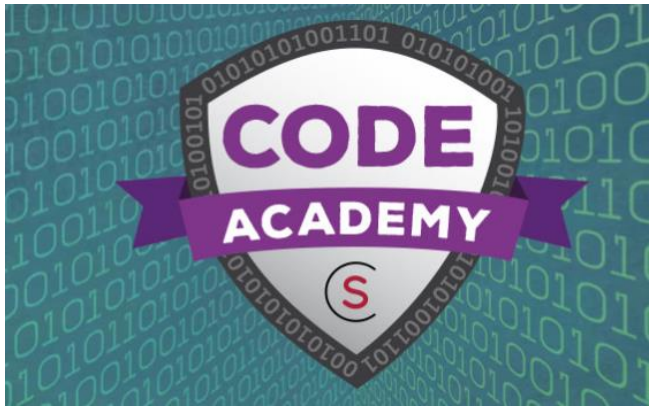



# Day 6: C# programming and .NET

CODE BOOTCAMP VAASA 2021

JANI JÄRVINEN, INTERTECHNO TRAINING OY

# Object oriented programming

# Object oriented programming concepts

- Object oriented programming (OOP for short) is a key concept in programming, and required for programming in languages like C++, Java, C#, Visual Basic .NET, Scala and Kotlin
- Important terminology
  - Class, instance
  - Method, property, field
  - Inheritance
  - Access modifiers
  - Interfaces
- Special notes on C#
  - All code must exist inside a class
  - Does not support multiple inheritance like C++ does

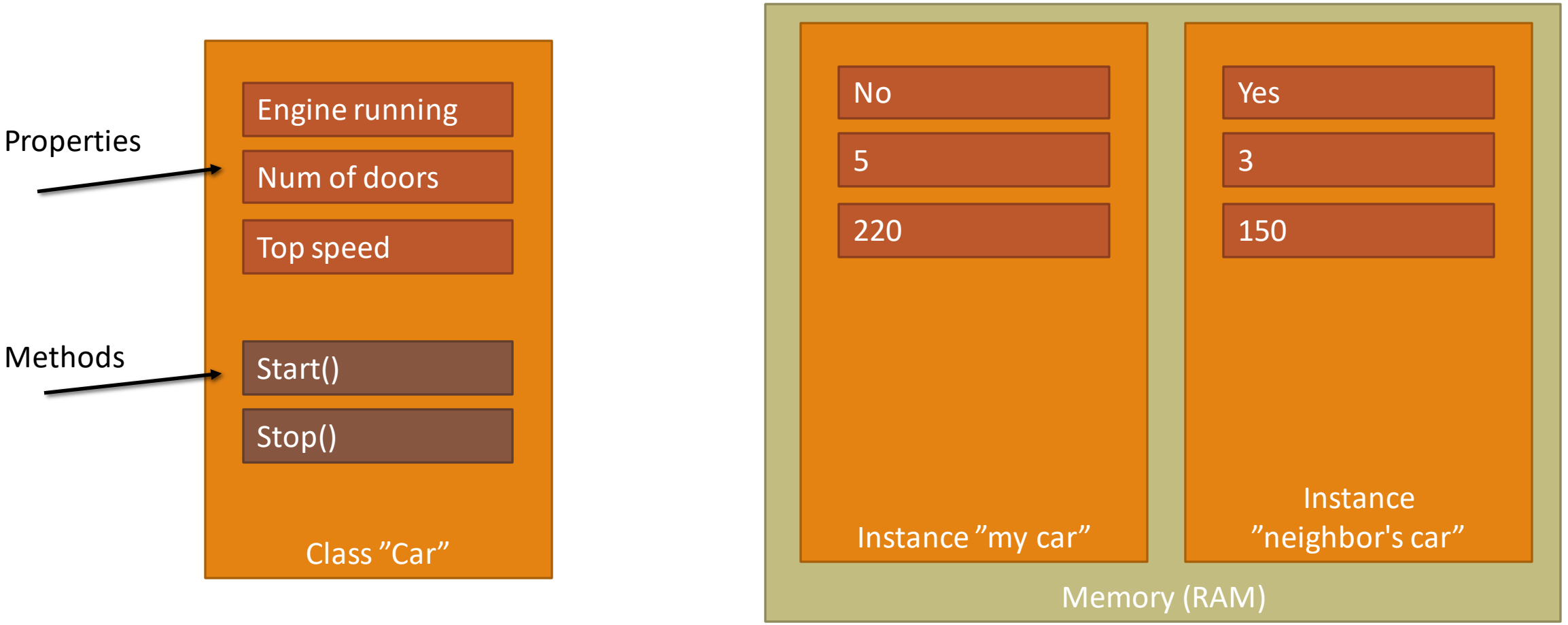
# Classes

- One of the key concepts in object oriented programming
- Defines an application concept in an abstract manner
- Classes model the outside world, and can be for example a “car”, “customer”, “order”, “employee”, “maintenance item” and the like
- Each class has some *functionality* (methods) and some *data*
  - Furthermore, each class has a *public interface* that can be seen outside, and a *private implementation*
- As long as a classes public interface stays the same, the internal implementation can change
  - A class can inherit functionality from another classes, this is called inheritance
  - A classes ability to hide its internal implementation is called encapsulation

# Classes and objects in .NET

- All classes in .NET (and thus, also C#) inherit from a class called System.Object
- A class is usually defined in a single source code file, but it can also span multiple files
  - Partial classes and partial methods
- C# example
  - ```
    public class Customer {
        …
    }
    ```
  - `Customer cust = new Customer();`
- The *this* keywords
  - Points to ”the current object”
    ```

# Class and instance (aka object)



# What does a class contain?

- In C#, a class can contains the following
  - Functionality: *methods*
  - Data: *properties*
  - Data: *fields*
  - Responds to: *events*
- Normally, properties and fields are instance related, but they can also be static
  - A static property or field is class related, meaning that all instances of the class share the same values
- C# 9.0 also supports records, and C# 10.0 supports record structs

# Summary and exercises

# Exercises

1. Investigate how much memory does a .NET application consume. How can you monitor such things?
2. In many programming languages, there is something called a runtime. In .NET development, there is a technology called CLR. What is this CLR, and what does it do?
3. In which operating systems can you run applications written with C#?
4. How do you know which .NET version does your application use? Bonus question: how do you know which C# language version does your application use?

# Coding

1. Define a variable in C# named “strangeValue” that is of type float. Set the variable’s value to be 9999999.4999999999. Then, use Console.WriteLine command to display the value on the screen. Which value is displayed? Investigate why this happens.
2. Find ways to generate random numbers in C#. Examine how these random numbers are generated. Would the method you found be suitable in generating secure passwords or tokens? Why, or why not?
3. Sometimes, applications you write will generate errors. In .NET, these runtime errors are called exceptions. Examine different ways to manage these exceptions.