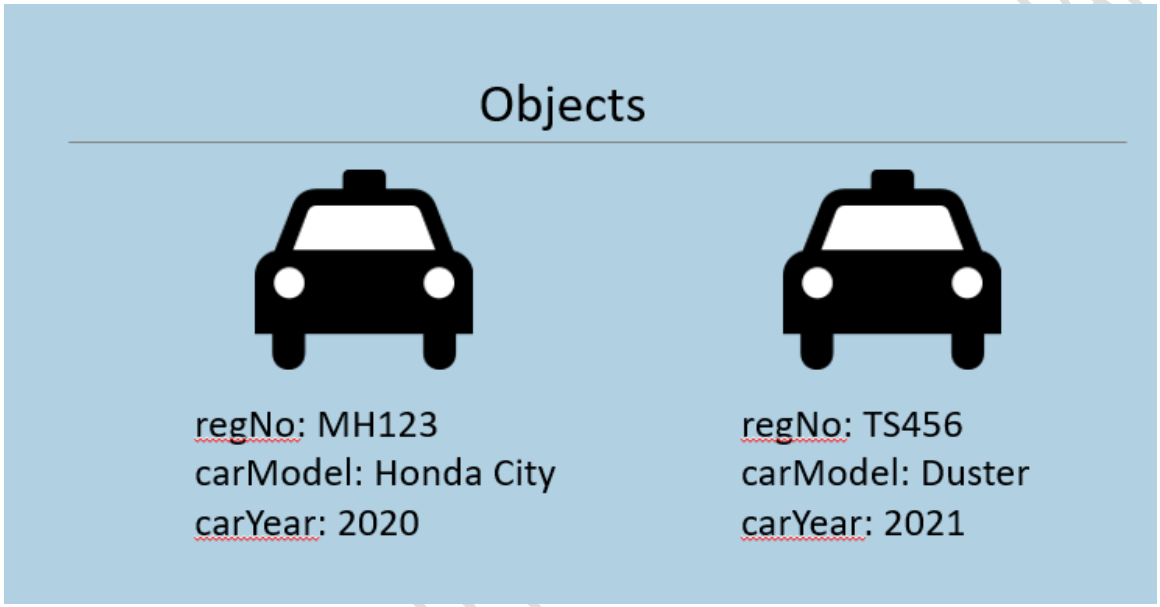
C# - Ultimate Guide - Beginner to Advanced | Master class

Section 5 - Fields

Fields

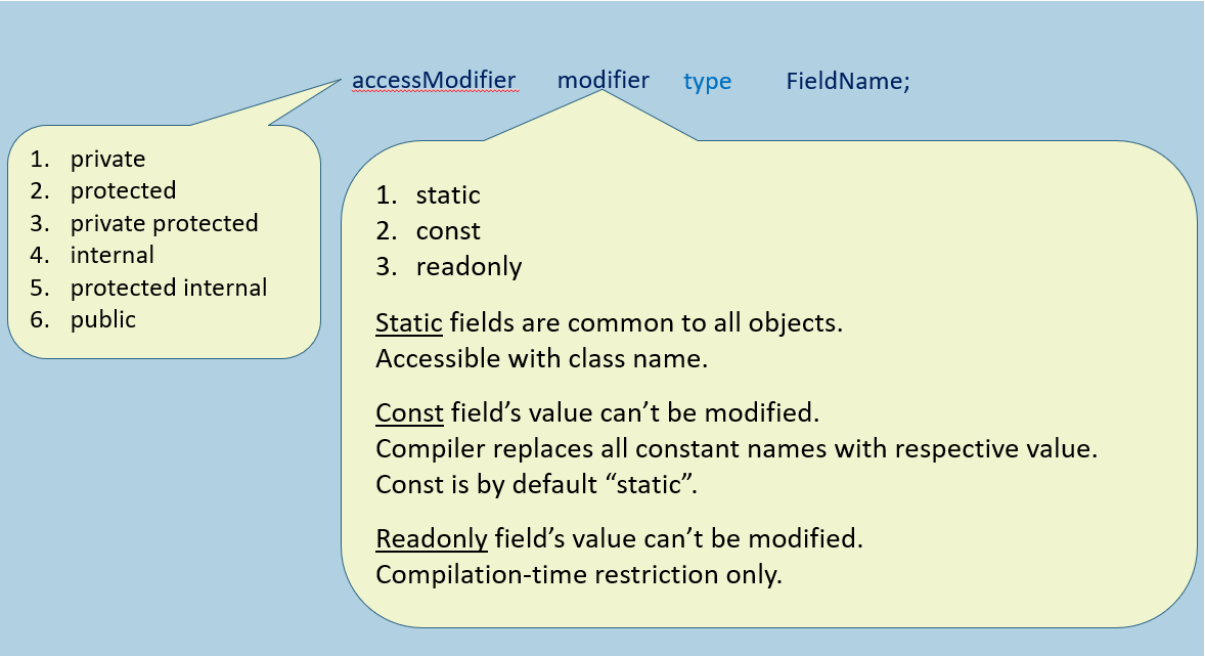
Variables that are declared in the class; stored in the objects.

Isolated for each object.




```
class Car
{
  string regNo;
  string carModel;
  int carYear;
}
```

Syntax of Field



Access Modifiers of Fields

Access Modifier (a.k.a. "Access Specifier" or "Visibility Modifier) specifies the accessibility of fields, where the fields can be accessible; they provide security for the fields.

Access Modifier	In the same class	In the child classes at the same assembly	In the other classes at the same assembly	Child classes at other assembly	Other classes at other assembly
private	Yes	No	No	No	No
protected	Yes	Yes	No	Yes	No
private protected	Yes	Yes	No	No	No
internal	Yes	Yes	Yes	No	No
protected internal	Yes	Yes	Yes	Yes	No
public	Yes	Yes	Yes	Yes	Yes

Static Fields

Static fields are store outside the object.

Static fields are common to all objects of a class.

Class Memory in Heap

bankName: Bank of Dummyland

Objects in Heap

object 1:

accountNumber: 1001

accountHolderName: Scott

currentBalance: 5000

object 2:

accountNumber: 1002

accountHolderName: Bob

currentBalance: 6000

```
class BankAccount
{
  long accountNumber;
  string accountHolderName;
  double currentBalance;
  static string bankName;
}
```

Instance Fields (vs) Static Fields

Storage:

Instance Fields: Stored in Objects

Static Fields: Stored in Class's memory.

Related to:

Instance Fields: Represents data related to objects.

Static Fields: Represents common data that belongs to all objects.

Declaration:

Instance Fields: Declared without "static" keyword. Syntax: type fieldName;

Static Fields: Declared with "static" keyword. Syntax: static type fieldName;

Accessible with:

Instance Fields: Accessible with object (through reference variable).

Static Fields: Accessible with class name only (not with object).

When memory gets allocated:

Instance Fields: Allocated separately for each object, because instance fields are stored "inside" the objects.

Static Fields: Allocated only once for the entire program; i.e. when the class is used for the first time while executing the program.

Constant Fields

Constant Fields are like static fields, that are common to all objects of the class.

We can't change the value of constant field.

Constant Fields are accessible with class name [not with object].

Constant Fields are not stored in the object; will not be stored anywhere.

Constant Fields will be replaced with its value, while compilation; so it will not be stored anywhere in memory.

Constant Fields must be initialized, in line with declaration (with a literal value only).

Constants can also be declared as 'local constants' (in a method).

AccessModifier const type FieldName = value;

Readonly Fields

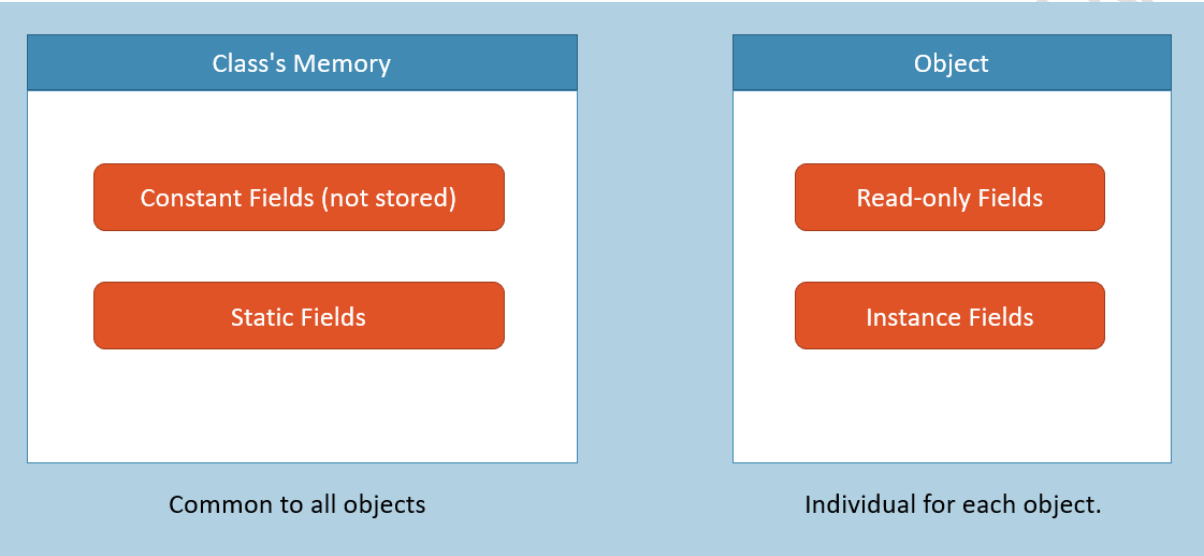
Readonly Fields are like instance fields, that is stored in every object, individually.

We can't change the value of readonly field.

Readonly Fields are accessible with reference variable [with object].

Readonly Fields must be initialized, either "in-line with declaration" [or] "in the constructor".

AccessModifier readonly DataType FieldName = value;



Key Points to Remember

- Fields are variables that are declared in the class; but stored in objects.
- Access modifiers of fields: private, protected, private protected, internal, protected internal, public
- Modifiers of fields: static, const, readonly
- Instance fields are individual for each object; Static fields are common (one-time) for all objects.
- Constants must be initialized along with declaration; Readonly fields must be initialized either 'along with declaration' or in 'instance constructor'.