

Section 5: Fields

09 July 2024 12:23



Harsha Vardhan

- › UI Expert
- › .NET Expert
- › Module Lead
- › Corporate Instructor



Fields

Agenda

- 1 Understanding Fields
What is field and how it gets stored.
- 2 Syntax of Field
How to create field?
Understanding access modifiers, modifiers of fields etc.
- 3 Access Modifiers of Fields
private, protected, private protected, internal, protected internal, public
- 4 Static Fields
Instance fields (vs) Static fields

Agenda

- 5 Constant Fields
What is constant?
Where and how constants are stored?
- 6 Readonly Fields
Features of Readonly fields
Constants (vs) Readonly Fields

What

- Variables that are declared in the class; stored in the object.
- Isolated for each object.

Objects

regNo: MHI23
carModel: Honda City
carYear: 2020



regNo: TS456
carModel: Duster
carYear: 2021

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

Syntax

accessModifier modifier type FieldName;

- private
- protected
- private protected
- internal
- protected internal
- public

- static
- const
- readonly

Static fields are common to all objects.
Accessible with class name.

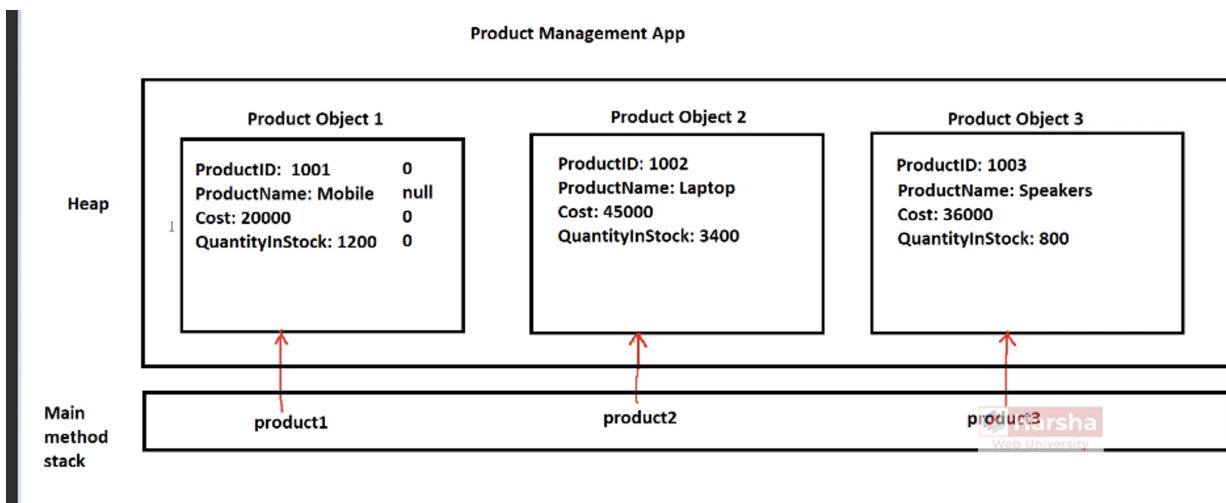
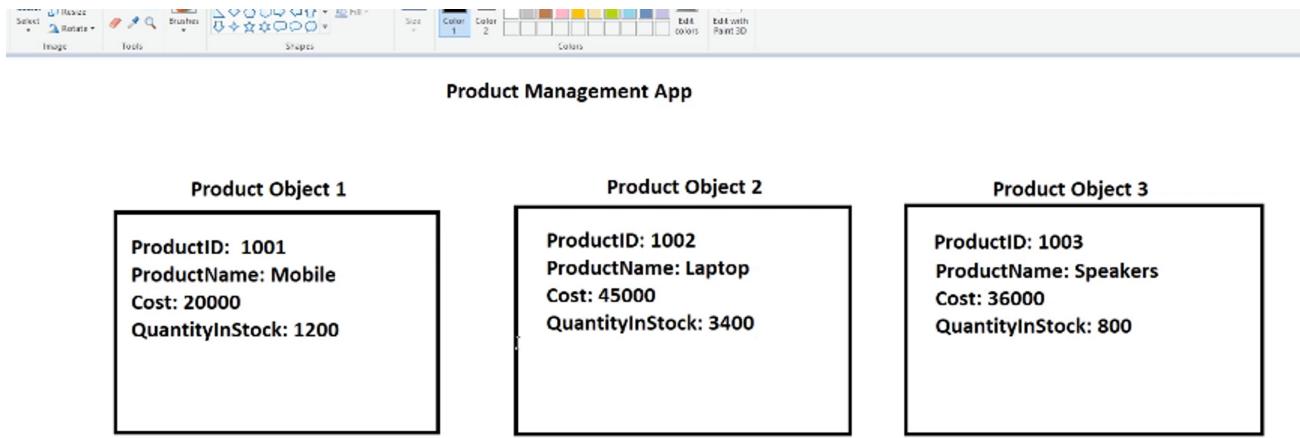
Const field's value can't be modified.
Compiler replaces all constant names with respective value.
Const is by default "static".

Readonly field's value can't be modified.
Compilation-time restriction only.

optional

The Default Access Modifier of a field is Private.

The Default Access Modifier of a class is Internal.



Access Modifier of Fields

Syntax of Field → **Next: Access Modifiers of Fields**

Syntax

accessModifier modifier type FieldName;

- 1. private
- 2. protected
- 3. private protected
- 4. internal
- 5. protected internal
- 6. public

1. static
2. const
3. readonly

Static fields are common to all objects.
Accessible with class name.

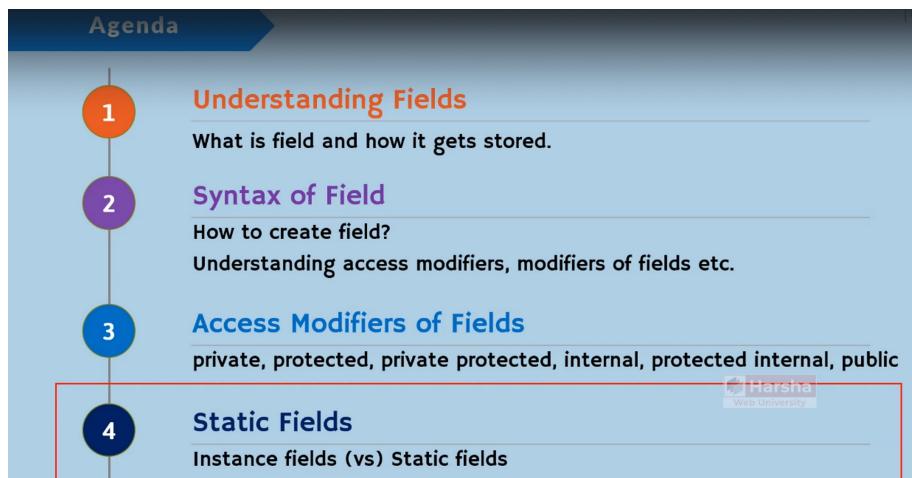
Const field's value can't be modified.
Compiler replaces all constant names with respective value.
Const is by default "static".

Readonly field's value can't be modified.
Compilation-time restriction only.

Class Library file are compiled as DLL, that DLL is called as assembly.
Console Application is compiled as exe, that exe is called as assembly.

What	Access Modifiers of Fields				
	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

Next: Instance Fields (vs) Static Fields

What

- › Static fields are stored outside the object.
- › Static fields are common to all objects of a class.

Class Memory in Heap

bankName: Bank of Dummyland

Objects in Heap

accountNumber: 1001
accountHolderName: Scott
currentBalance: 5000

accountNumber: 1002
accountHolderName: Bob
currentBalance: 6000

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

Instance Fields (vs) Static Fields

Next: Constant Fields

Instance Fields

Static Fields

Storage

- › Stored in Objects

- › Stored in Class's memory.

Related to

- › Represents data related to objects.

- › Represents common data that belongs to all objects.

Declaration

- › Declared without "static" keyword.
- › Syntax: type fieldName;

- › Declared with "static" keyword.

› Syntax: static type fieldName;

Accessible with

- › Accessible with object (through reference variable).

- › Accessible with class name only (not with object).

Instance Fields (vs) Static Fields

Next: Constant Fields

Instance Fields

Static Fields

When memory gets allocated

- › Allocated separately for each object, because instance fields are stored "inside" the objects.

- › Allocated only once for the entire program; i.e. when the class is used for the first time while executing the program.

Constant Field

Agenda

5

Constant Fields

What is constant?

Where and how constants are stored?

6

Readonly Fields

Features of Readonly fields

Constants (vs) Readonly Fields

Constant Fields

Next: Readonly Fields

What

- > 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

Agenda

5

Constant Fields

What is constant?

Where and how constants are stored?

6

Readonly Fields

Features of Readonly fields

Constants (vs) Readonly Fields

Readonly Fields

| Har

What

- › 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".

 Harsha
Web University

AccessModifier **readonly** DataType FieldName = value;

Registration no of a student is an example of readonly field.

Key Points to Remember

| Harsha

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'.

 Harsha
Web University