



Harsha Vardhan

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

Introduction to OOP, Classes, Objects

Agenda

- 1 **Introducing OOP** ←
- 2 **Understanding Objects and Classes**
- 3 **Understanding Methods**
- 4 **Class & Object Association**
- 5 **Creating Classes and Objects**

What is OOP, Goal of OOP.

What are Objects and its features? What are Classes and its features.
Understand objects in real-world programming.

What is method; Understanding parameters and return.

How objects are associated to classes in memory?

Understanding steps and rules involved in creating classes and objects.

Introducing OOP Next: Objects

What

- › Object Oriented Programming.
- › Programming Model for Scalable Applications.



- › Used in most popular languages such as Java, Python, JavaScript, C++ etc.
- › Goal of OOP is to group-up some data and its operations as a single unit called "Object".

Objects Next: Classes

Object

- › Object is a small unit (entity) in the program that represents a real-world person or thing.
Ex: You, Your laptop
- › Any physical thing can be considered as object.
- › Object is instance (example) of "class".
- › Object stores a set of fields (details about object).

 regNo: MHI23 carModel: Honda City carYear: 2020	 regNo: TS456 carModel: Duster carYear: 2021	 regNo: TS789 carModel: Swift carYear: 2019
--	--	---

Classes Next: Methods

Class

- > Class is a model of objects.
- > Class (a.k.a "type") represents structure (list of fields and methods) of data that you want to store in similar objects.
- > Class isn't collection of objects.
- > Objects are created based on "Class".

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

Methods Next: Object & Class Association

Method

- > Method is a collection of statements to perform certain operation (process or work), such as performing some calculation, displaying some output, checking some conditions etc.
- > Method should be a member (part) of class.

```
class Car
{
    int calculateEmi( int carPrice, int noOfMonths, int interestRate )
    {
        //do calculation here
        return (emi);
    }
}
```

Object does not store methods. It just stores fields. Objects are associated to the class's methods.

Object & Class Association Next: Creating Class

Heap

Object stores fields.
Object associates with all methods of its class.
Means, object can call methods of its class.
Class declares list of fields; defines list of methods.

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

    int calculateEmi( int carPrice, int noOfMonths, int interestRate )
    {
        ...
    }
}
```

accessModifier class ClassName

```
{
    Fields
    Methods
    Constructors
    Properties
    Events
    Destructors
}
```

Harsha Web University

1. internal
2. public

Internal class is accessible within the same assembly.

Public class is accessible in the same assembly and also in other assemblies.

non object.

```
accessModifier class ClassName
{
    Fields
    Methods
    Constructors
    Properties
    Events
    Destructors
}
```

Harsha Web University



An Assembly is a project.

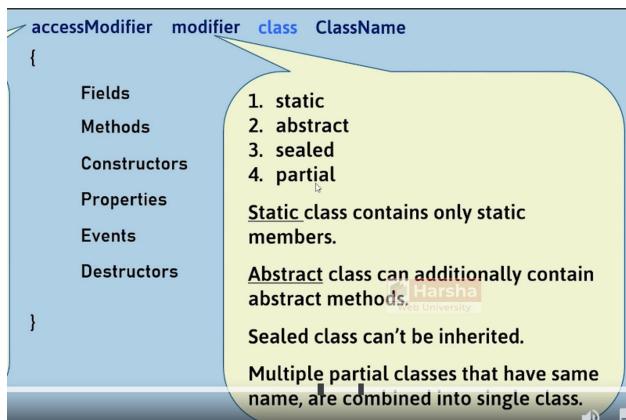
A solution is a collection of Projects.

The Compiled Source Code of a project is called as Assembly.

You created a console application with a set of files, all the code that is present in all the files of the console application will be compiled as an 'exe' file.

But in case of 'Class Library' type of project, all the files are compiled into a 'dll' file. So these 'exe' or 'dll' files are called as Assembly because they contain the compiled source code of the project in the form of 'Intermediate Level Language Code'.

The Default Access modifier is Internal.



Fields: The fields are used to store the details about the object.]

Methods: Methods are used to manipulated.

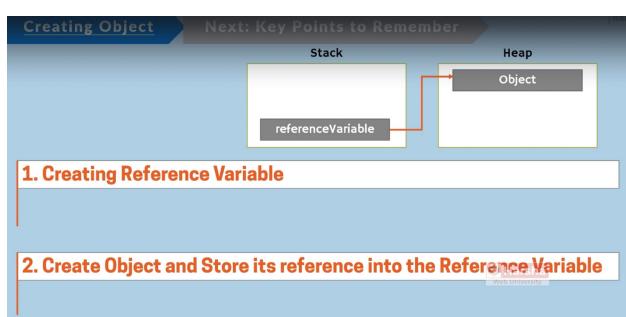
Constructor: are used to initialize the fields.

Properties: are used to set or get the values into the private fields.

Events: are used to raise the notification to other classes.

Destructors: are used to clear unmanaged resources.

The 'Class Library' is a collection of classes that will be compiled as a 'dll' (Dynamic Link Library) file



Objects are by default nameless. Object is a collection of fields that is inside in the heap.

1. Creating Reference Variable

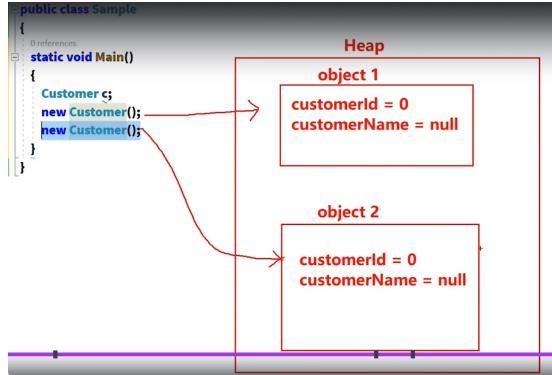
```
ClassName referenceVariable;  
Customer c;
```

1. Creating Reference Variable

```
ClassName referenceVariable;
```

2. Create Object and Store its reference into the Reference Variable

```
new ClassName();
```



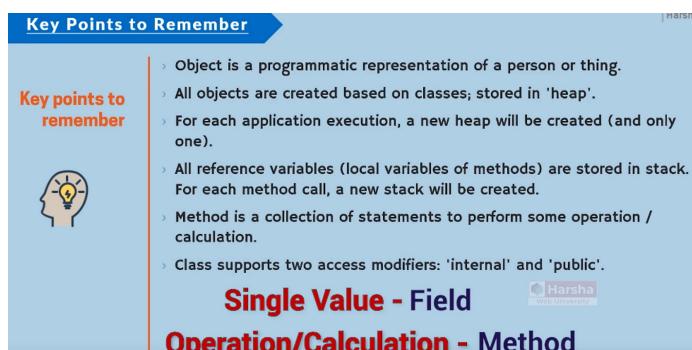
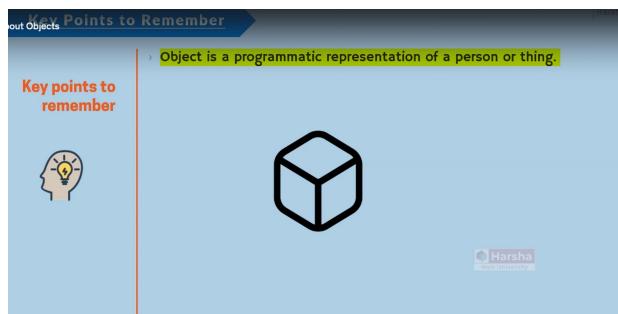
Class is a model of Object.

Fields is a individual detail about the Object.

Reference variables are used to store the reference of object.

All the objects are stored inside heap only.

Reference variables are local variables.



> Class supports four modifiers: 'static', 'abstract', 'sealed' and 'partial'.

> Objects stores actual data (group of fields) & can access methods of class.

> A reference variable stores address of an (only one) object.

