Introduction to OOP

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

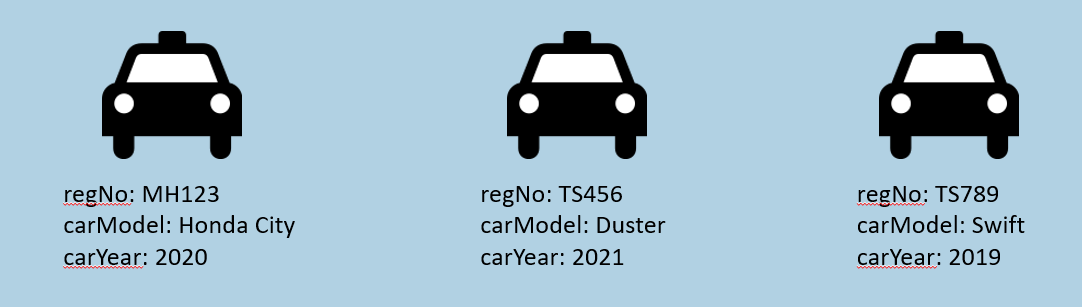
Object is a small unit (entity) in the program that represents a real-world person or thing.

Eg: 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).



Classes

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".

Eg:

1. class Car
2. {
3. string regNo;
4. string carModel;
5. int carYear;
6. }

Methods

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.

The code statements are not allowed outside the class; they are allowed inside the method only.

Eg:

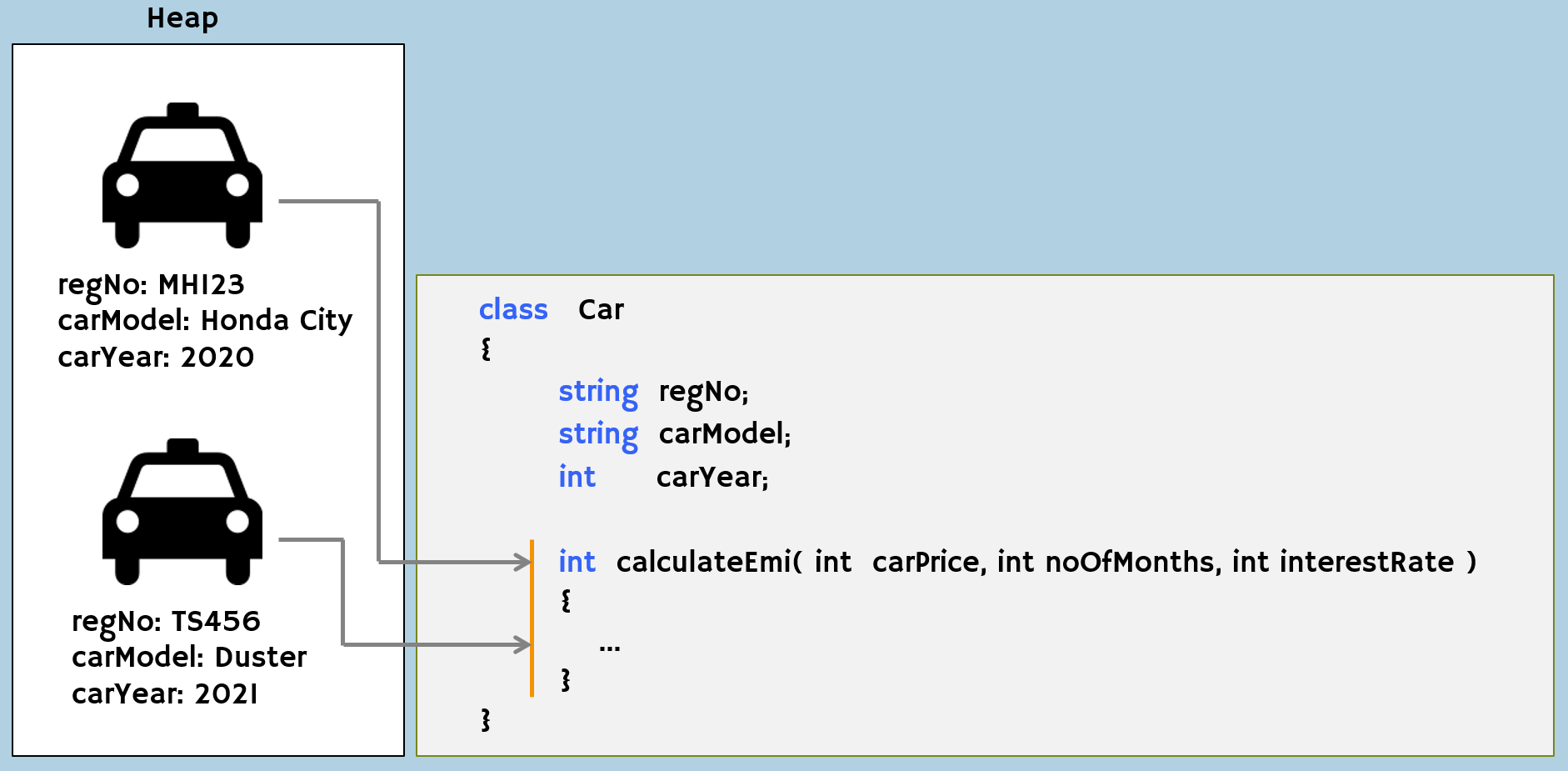
1. class Car
2. {
3. int calculateEmi( int carPrice, int noOfMonths, int interestRate )
4. {
5. //do calculation here
6. return (emi);
7. }
8. }

Object & Class Association

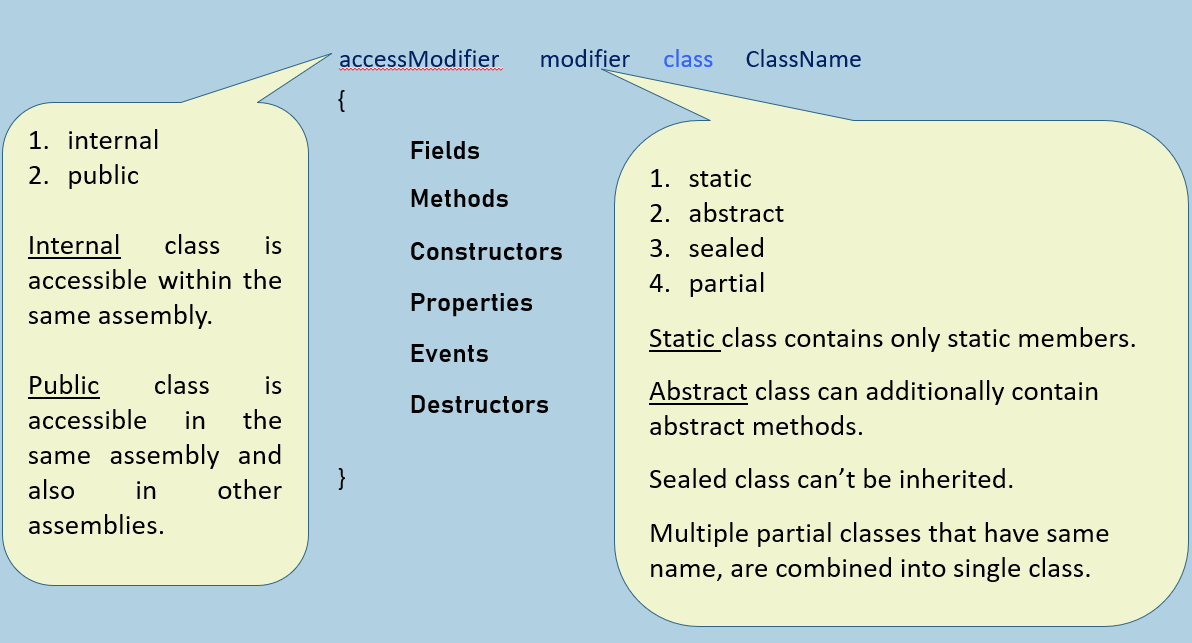
Object stores fields.

Object associates with all methods of its class. Means, object can call methods of its class.

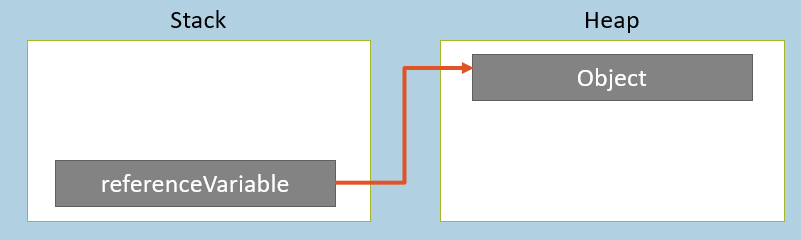
Class declares list of fields; defines list of methods.



**Creating Class**



**Creating Object**



1. Creating Reference Variable

   ClassName referenceVariable;

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

     referenceVariableName = new ClassName( );

**Key Points to Remember**

* Object is a programmatic representation of a person or thing.
* All objects are created based on classes; stored in 'heap'.
* For each application execution, a new heap will be created (and only one).
* All reference variables (local variables of methods) are stored in stack. For each method call, a new stack will be created.
* Method is a collection of statements to perform some operation / calculation.
* Class supports two access modifiers: 'internal' and 'public'.
* 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.