Modifiers (Continued)

static Non-Access Modifier

- Java classes cannot be specified with 'static' non-access modifier - Demonstrate
- Variables declared directly inside the class but outside the methods and are specified with 'static' modifier are known as static variables
- Memory allocated to the static variables is different from the memory allocated to the instance variables - view here
- static variables needs to be accessed with the help of Class name, as they belong to the Class memory - View here
- static variables are generally used to store common data, where as Object variables/Instance variables are used to store Object specific data.
  - wheels variable can be used as a static variable/class level variable as it has common data i.e. wheels count is 4 for all the cars in the market
  - Where as cost variable cannot be used as a static variable as its value changes from car to car, hence we use it as an Object variable/Instance variable.
- static can also be used with methods
- static can only access static stuff
  - You have to create object to overcome this

final Non-Access Modifier

- The value of the variable cannot be changed on specifying it with final non-access modifier - Demonstrate here
- final modifier specified classes cannot be inherited/extended by other classes - Demonstrate here
- final modifier specified methods in a class cannot be overridden by its sub-classes - Demonstrate here

abstract Non-Access Modifier

- variables cannot be specified with 'abstract' non-access modifier - Demonstrate
- On specifying a method with abstract modifier, we can just declare the method without implementing it - Demonstrate here
- Classes having at-least one abstract specified method must be specified as abstract
- Sub-Class inheriting the Super-Class needs to implement the abstract specified methods in Super-Class - Demonstrate here
  - Purpose of abstract methods - Used when the super-class dont have to implement everything, and when the sub-classes inheriting the super-class needs to implement them.
- Objects cant be created for abstract classes, we have to create a Sub-Class and access its variables/methods using Sub-Class object reference - Demonstrate here