* To perform initialization of object constructor is responsible
* Constructor is used to initialize all global variables.
* *By default JVM provide Default constructor.(No-arg constructor)*
* *It only provides default constructor if we havent provided any constructor.*
* *Default constructor always uses the same access modifier same as Class modifier.(public, <default>)*
* *It cannot be private and protected because class cannot be private and protected.*
* *Default constructor is a no-arg call to super class constructor*
* *Constructor overloading is possible.*
* *We can have n number of constructors with different arguments*

***Types of Constructors:-***

1. ***Default Constructors. (No-args Constructor.)***
2. ***Parameterised Constructor***

***Rule:-***

1. ***Name of class and constructor should be same.***
2. ***Constructors don’t need any return type as they are invoked on object creation.***

*(But if we provide return type to a constructor, then JVM will treat that constructor as method)*

1. *Modifiers allowed for constructors:* ***public, <default>, private, protected***

*(Singleton classes constructors are declared as private)*

1. *Cocrete class , abstract classes can have constructors, but interface can never have constructors.(Because abstract classes can have instance variables, but in case of interface all variables are public static final by default)*
2. *Whenever we are writing any parameterised constructor in parent class and then extending it to child class, then its mandatory that we should write Default constructor in Parent class as well.*
3. *If parent class constructor throws any Checked Exception then compulsorily child class constructor should throw the same exception or parent Exception, or else we will get an error.*

***Super() & this():-***

* *Super(), this() compulsorily should always be the* ***first line of constructor****.*
* *We can use either super(), this(),* ***we cannot use both at same time****, as they should always be wirtten on first line.*
* *We can use super(), this ()* ***only inside constructors.***
* *This(), this* ***refers to current class*** *object.*
* *Super(), super* ***refers to parent class*** *object*
* *If we want to call constructors of same class then use* ***this().***
* *If we want to call constructor of parent class use* ***super()***
* *Super(), this() are used to call constructors.*
* *Super, this are used to call instance variables of parent , child respectively. (Super.name),(this.name)*
* *We cannot use super and this inside static areas/methods/blocks because static refers to class level and super, this refers to object level*
* *Since constructors are tightly bound to their class, they do not get inherited by subclasses.*

***Thus constructors cant be overrided and inherited.***