|  |  |
| --- | --- |
| **Constructors** | **Method** |
| A constructor is used to initialize the instance variable of a class. | A method is used for any general purpose processing or calculation. |
| A Constructors name and class name should be same. | A method name and class name may be same or different. |
| A Constructor is called at the time of creating the objects. | A method can be called after creating the objects. |
| A Constructor is called only one per object. | A method can be called several times on the object. |
| A Constructor is called and executed automatically. | A method is executed only when we call it. |
| A constructor never containing any return type. | A method must have a return type. |

**Difference between Method overloading and Method overriding**

|  |  |
| --- | --- |
| **Method Overloading** | **Method overriding** |
| Writing two or more methods with the same name but with different parameter is called method overloading. | Writing two or more methods with the same name and same signature is called method overriding. |
| Method overloading is done in the same class. | Method overriding is done in super and sub classes. |
| In method overloading, method return type can be same or different. | In method overriding, method return type should also be same. |
| JVM decides which method is called depending on the difference in method signature. | JVM decides which method is called depending on the data type (class) of the object used to call the method. |
| Method overloading is done when the programmer wants to extend the already available features. | Method overriding is done when the programmer wants to provide a different implementation (body) for the same feature. |
| Method overloading is code refinement. Same method is refined to perform the different task. | Method overriding is code replacement. The sub class method overrides (replaces) the super class method. |

**Difference between this and super keyword:**

|  |  |
| --- | --- |
| **This** | **Super** |
| It can be used to differentiate the local variable and instance variable. | Super can be differentiating the super class variable and sub class variable. |
| Will be participating function chaining, in function chain method must return object. | Whenever super class method name, sub class method names both are same to differentiating them using super. |
| Constructor chaining that to same class constructor will be called up one after another. | Super can be calling super class constructor from sub class by using super we can able to call the required constructor from super class. |
| This class always is used within the same class only. | Super can be used always within the sub class |
| While calling same class constructor this keyword must be first line within the constructor. | In this case also within the sub class super keyword must be in the first line. |
| It can be placed anywhere within the method as part of method and variables. | Super can also be placed anywhere within the method. |

**Difference between abstract class and interface:-**

|  |  |
| --- | --- |
| An abstract class contain all the abstract methods or all the concrete method or both. | It is one type of class which is containing abstract methods only. |
| Abstract class can have variable with any access specifier except private. | Interface variable are by default public, static and final. |
| Abstract class can have methods with any access specifier except private. | Interface methods are by default public so that at the time of overriding, compulsory we should write at public access specifier. |
| Abstract keyword is required in order to declare abstract method. | In interface abstract keyword is not required because by default interface methods are abstract methods. |
| Multiple inheritance is not possible. | Multiple inheritance is possible. |