

Abstract class

✓ abstract method
abstract class

```
class Test
```

```
{  
    p.s.v.main()  
}
```

✓ Super s1;

✗ s1 = new Super();

✓ Sub s2 = new Sub();

```
}
```

```
abstract class Super
```

```
{  
    {super()  
    {s.o.p("Super");  
}
```

```
void meth1()  
{  
    s.o.p("Super meth1");  
}
```

```
abstract void meth2();  
}
```

```
class Sub extends Super
```

```
{  
    void meth2()  
}
```

```
{  
    s.o.p("Sub meth2");  
}
```

```
}
```

--> There are two types of classes Abstract class and Concrete class.

--> If **abstract** keyword is used before the class then it is an Abstract Class if nothing is written before class then it is a Concrete class.

--> "Abstraction" is a process of hiding the implementation or way of managing complexity by hiding unnecessary details and showing only functionality to the user.

● Object of an Abstract class cannot be created but object of Concrete class can be created.

● Reference of abstract class is allowed.

--> "Method" which is not having a body is known as Abstract method, the method must be declared as abstract. Mtlb hma "method" ka andar kuch bhi nahi likhna hai.

--> A class is "**abstract class**" if at least one of the methods is abstract.

--> If any "**sub-class**" inherits abstract class then that "**sub-class**" also becomes "abstract class"; lekin agar ham "sub-class" ko "**concrete class**" mai change karna chahta hai to hma "undefined method of super-class" ko override karna hoga.

Abstract class

Do and Dont

1. `abstract class Test`
`{`
`=`
`}`

✓ `Test t;`

✗ `t = new Test();`

2. ✗ `abstract final class Test`
`{`
`-`
`}`

3. ✗ `abstract static class Test`
`{`
`-`
`}`

4. `abstract class Test`
`{`
`✗ final abstract void display();`
`}`

5. `abstract class Test1`
`{`
`abstract void display();`
`}`

• `class Test2 extends Test1`
`{`

`✗ void display()`
`{`
`=`
`}`
`}`

--> An Abstract class cannot be final because if it is made final then it cannot be extended whereas abstract class is meant for inheritance.

--> An Abstract method cannot be final because if it made final then it cannot be overridden whereas Abstract method is meant for overriding.

--> Abstract Class and method can neither be final nor static.

--> A Sub class must override an abstract method or else it will become abstract class.