Abstract class in Java -

> A class which is declared as abstract is known as an abstract class. It can have abstract and non-abstract methods.

> It needs to be extended and its abstract method implemented.

* > It cannot be instantiated.

Points to remember :

> An abstract class must be declared with an abstract keyword.

> It can have abstract and non-abstract methods.

> It cannot be instantiated.

> It can have constructors and static methods also.

> It can have final methods which will force the subclass not to change the body of the method.

Scanned by CamScanner

#

> an instance of an abstract class cannot be created, we can have references of abstract type though.

```
abstract class Base{
      abstract void fun();
 }
class Derived extends Base
{
      void fun()
    { sopn("Derived fun called"); }
 }
class Main{
    public static void main(String args[])
   {
        Base b = new Derived();
        b.fun();
   }
}
```

> An abstract class can contain constructors in Java. And a constructor abstract class is called when an instance of a inherited class is created.

Scanned by CamScanner

> We can have abstract class without any abstract method. This allows us to create classes that cannot be instantiated, but can only be inherited.

```
abstract class Base
{
        void fun()
        { sopn(" Base fun() called"); }
}
class Derived extends Base{ }
{lass main{
        p s vm(String args[])
        {
            Derived d=new Derived();
            d.fun();
        }
}
```

> Abstract class can also have final methods (methods that cannot be overriden).

```
abstract class Base{
 final void fun{

        SOPn(" final method called");
 }
```

Scanned by CamScanner

```
class Derived extends Base{}

class Main{
      public static void main(String args[])
      {
            Base b= new Derived();
            b.fun();
        }
  }
```

Scanned by CamScanner