Abstraction in Jav0061

**Abstraction** is a process of hiding the implementation details and showing only functionality to the user.

Student

studentId🡪 visible – non abstract---->normal method

studentMark🡪 hide 🡪 abstract--->abstract method

studentStaff🡪 hide🡪 abstract

Staff extends Student

studentId

studentMark{

}

studentStaff{

}

### Ways to achieve Abstraction

There are two ways to achieve abstraction in java

* Partial Abstraction or Abstract Class
* Fully Abstraction or Interface.

#### **Points to Remember**

* Supports Abstract and Non-Abstract method--->partial abstraction
* We cannot create an object for Abstract Class because there is no implementation part.
* By using “extends” keywords we can access Abstract Class.
* public **abstract** Keyword is Mandatory.
* No implementation Part only Signature part.

**Example of abstract class**

1. **public abstract** **class** Classname{

}

**Example of abstract method**

1. public abstract void methodnName();//no method body in abstract
2. public void methodnName(){ }

Interface:

An **interface in Java** is a blueprint of a class.

* It supports only Abstract Method
* We cannot create an object for interface because there is no implementation part
* By using “**implements**” keyword we can access Interface
* By interface, we can support the functionality of **multiple inheritance**.

**Example of Interface class**

1. **Public interface**  Classname{

}

**Example of abstract method**

1. public abstract void methodnName();//no method body in abstract

OR

1. void methodnName();//no method body in abstract

