### \*\* Abstraction \*\*

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

There are two ways to achieve abstraction in java

1. Abstract class
2. Interface

­­­­

### Abstract class :

1. A class which is declared as abstract keyword is known as an **abstract class**.
2. Abstract class can have abstract and non-abstract methods.
3. To use the Abstract class, we need to inherit the abstract class.
4. We can’t create Object of the Abstract class.
5. Abstract class can have [constructors](https://www.javatpoint.com/java-constructor) and static methods also.
6. Abstract class can have final methods which will force the subclass not to change the body of the method.

### Abstract Method :-

### A method declared using the abstract keyword within abstract class

### and does not have definition is known as an abstract method.

1. To use the Abstract method, it is compulsory to inherit the class.
2. Sub class Object can use super class Methods.
3. If there is an abstract method in a class, that class must be abstract.
4. If you are extending an abstract class that has an abstract method, you must either provide the implementation(Overriding) of the method or make this class abstract.

abstract class A

**{**

**}**

class Q01\_Abstraction

**{**

    public static void main**(**String args**[])**

**{**

**}**

**}**

Compile And Run

abstract class A

**{**

    void show**()**

**{**

        System**.**out**.**println**(**"Abstract class"**);**

**}**

**}**

class Q02\_Abstraction

**{**

    public static void main**(**String args**[])**

**{**

        A a **=** **new** A**();**

**}**

**}**

error: A is abstract; cannot be instantiated

    A a = new A();

          ^

abstract class A

**{**

    void show**()**

**{**

        System**.**out**.**println**(**"Abstract class"**);**

**}**

**}**

class B **extends** A

**{**

**}**

class Q03\_Abstraction

**{**

    public static void main**(**String args**[])**

**{**

        B b **=** **new** B**();**

        b**.**show**();**

**}**

**}**

Abstract class

class A

**{**

    abstract void show**();**

**}**

class Q04\_Abstraction\_Method

**{**

    public static void main**(**String args**[])**

**{**

**}**

**}**

error: A is not abstract and does not override abstract method show() in A

class A

^

abstract class A

**{**

    void show**();**

**}**

class Q05\_Abstraction\_Method

**{**

    public static void main**(**String args**[])**

**{**

        System**.**out**.**println**(**"Done"**);**

**}**

**}**

 error: missing method body, or declare abstract

        void show();

             ^

abstract class A

**{**

    abstract void show**();**

**}**

class Q06\_Abstraction\_Method

**{**

    public static void main**(**String args**[])**

**{**

        System**.**out**.**println**(**"Done"**);**

**}**

**}**

Done

abstract class A

**{**

    abstract void show**()**

**{**

        System**.**out**.**println**(**"Abstract Class"**);**

**}**

**}**

class Q07\_Abstraction\_Method

**{**

    public static void main**(**String args**[])**

**{**

        System**.**out**.**println**(**"Done"**);**

**}**

**}**

Q07\_Abstraction\_Method.java:3: error: abstract methods cannot have a body

        abstract void show()

                      ^

abstract class A

**{**

    abstract void show**();**

**}**

class B **extends** A

**{}**

class Q08\_Abstraction\_Method

**{**

    public static void main**(**String args**[])**

**{**

        System**.**out**.**println**(**"Done"**);**

**}**

**}**

B is not abstract and does not override abstract method show() in A

class B extends  A

^

abstract class A

**{**

    abstract void show**();**

**}**

abstract class B **extends** A

**{**

**}**

class Q09\_Abstraction\_Method

**{**

    public static void main**(**String args**[])**

**{**

        System**.**out**.**println**(**"Done"**);**

**}**

**}**

Done

abstract class A

**{**

    abstract void show**();**

**}**

abstract class B **extends** A

**{**

**}**

class Q10\_Abstraction\_Method

**{**

    public static void main**(**String args**[])**

**{**

        B b **=** **new** B**();**

        System**.**out**.**println**(**"Done"**);**

**}**

**}**

 error: B is abstract; cannot be instantiated

                B b = new B();

                      ^

abstract class A

**{**

    abstract void show**();**

**}**

class B **extends** A

**{**

    void show**()**

**{**

        System**.**out**.**println**(**"Class B"**);**

**}**

**}**

class Q11\_Abstraction\_Method

**{**

    public static void main**(**String args**[])**

**{**

        B b **=** **new** B**();**

        b**.**show**();**

        System**.**out**.**println**(**"Done"**);**

**}**

**}**

Class B

Done

abstract class A

**{**

    abstract void show1**();**

    abstract void show2**();**

**}**

class B **extends** A

**{**

    void show1**()**

**{**

        System**.**out**.**println**(**"Class B"**);**

**}**

**}**

class Q12\_Abstraction\_Method

**{**

    public static void main**(**String args**[])**

**{**

        B b **=** **new** B**();**

        b**.**show1**();**

        System**.**out**.**println**(**"Done"**);**

**}**

**}**

error: B is not abstract and does not override abstract method show2() in A

class B extends A

^

abstract class A

**{**

    abstract void show1**();**

    abstract void show2**();**

**}**

class B **extends** A

**{**

    void show1**()**

**{**

        System**.**out**.**println**(**"Class B Show1"**);**

**}**

    void show2**()**

**{**

        System**.**out**.**println**(**"Class B Show2"**);**

**}**

**}**

class Q13\_Abstraction\_Method

**{**

    public static void main**(**String args**[])**

**{**

        B b **=** **new** B**();**

        b**.**show1**();**

        b**.**show2**();**

        System**.**out**.**println**(**"Done"**);**

**}**

**}**

Class B Show1

Class B Show2

Done

abstract class A

**{**

    abstract void show1**();**

    abstract void show2**();**

**}**

class B **extends** A

**{**

    void show1**()**

**{**

        System**.**out**.**println**(**"Class B Show1"**);**

**}**

**}**

class C **extends** B

**{**

**}**

class Q14\_Abstraction\_Method

**{**

    public static void main**(**String args**[])**

**{**

        B b **=** **new** B**();**

        b**.**show1**();**

        b**.**show2**();**

        System**.**out**.**println**(**"Done"**);**

**}**

**}**

error: B is not abstract and does not override abstract method show2() in A

class B extends A

^

abstract class A

**{**

    abstract void show1**();**

    abstract void show2**();**

**}**

class B **extends** A

**{**

    void show1**()**

**{**

        System**.**out**.**println**(**"Class B Show1"**);**

**}**

**}**

class C **extends** B

**{**

    void show2**()**

**{**

        System**.**out**.**println**(**"Class B Show2"**);**

**}**

**}**

class Q15\_Abstraction\_Method

**{**

    public static void main**(**String args**[])**

**{**

        B b **=** **new** B**();**

        b**.**show1**();**

        b**.**show2**();**

        System**.**out**.**println**(**"Done"**);**

**}**

**}**

error: B is not abstract and does not override abstract method show2() in A

class B extends A

^

abstract class A

**{**

    static void show1**()**

**{**

        System**.**out**.**println**(**"Class A"**);**

**}**

**}**

class Q16\_Abstraction\_Class\_Static\_Method

**{**

    public static void main**(**String args**[])**

**{**

        A**.**show1**();**

        System**.**out**.**println**(**"Done"**);**

**}**

**}**

Class A

Done

abstract class Q17\_Abstraction\_Class\_Main\_Method

**{**

    public static void main**(**String args**[])**

**{**

        System**.**out**.**println**(**"Done"**);**

**}**

**}**

Done

abstract class A

**{**

    A**()**

**{**

        System**.**out**.**println**(**"Abstract Class Constructor"**);**

**}**

**}**

class Q18\_Abstraction\_Class\_Constructor

**{**

    public static void main**(**String args**[])**

**{**

        System**.**out**.**println**(**"Done"**);**

**}**

**}**

Done

abstract class A

**{**

    A**()**

**{**

        System**.**out**.**println**(**"Abstract Class Constructor"**);**

**}**

**}**

class B **extends** A

**{}**

class Q19\_Abstraction\_Class\_Constructor

**{**

    public static void main**(**String args**[])**

**{**

        B b **=** **new** B**();**

        System**.**out**.**println**(**"Done"**);**

**}**

**}**

Abstract Class Constructor

Done

abstract class A

**{**

    A**()**

**{**

        System**.**out**.**println**(**"Abstract Class Constructor"**);**

**}**

    void show**()**

**{**

        System**.**out**.**println**(**"Abstract Class Show Method"**);**

**}**

**}**

class B **extends** A

**{}**

class Q20\_Abstraction\_Class\_Constructor

**{**

    public static void main**(**String args**[])**

**{**

        B b **=** **new** B**();**

        b**.**show**();**

        System**.**out**.**println**(**"Done"**);**

**}**

**}**

Abstract Class Constructor

Abstract Class Show Method

Done

**Abstract Class ka Reference variable sub class ke Object ko hold kar sakta he.**

abstract class A

**{**

    void show**()**

**{**

        System**.**out**.**println**(**"Abstract Class Show Method"**);**

**}**

**}**

class B **extends** A

**{}**

class Q21\_Abstraction\_Class\_Reference\_Variable

**{**

    public static void main**(**String args**[])**

**{**

        A a **=** **new** B**();**

        a**.**show**();**

        System**.**out**.**println**(**"Done"**);**

**}**

**}**

Abstract Class Show Method

Done