**Abstraction in Java** is the process of hiding the implementation details and only showing the essential details or features to the user. It allows one to focus on what an object does rather than how it does it. The unnecessary details are not displayed to the user.

**Key features of abstraction:**

* Abstraction hides the complex details and shows only essential features.
* Abstract classes may have **methods without implementation** and must be implemented by subclasses.
* By abstracting functionality, changes in the implementation do not affect the code that depends on the abstraction.

## **What is an Abstract Method?**

An abstract method is a method that is **declared** but not **defined** in a class. It acts as a placeholder for methods that must be implemented in subclasses.

## **What is an Abstract Class?**

In Java, a Class that contains at least one abstract method in it is called an abstract class and it is a class that **cannot be instantiated** on its own and is **meant to be subclassed**.

In Java, abstraction is achieved by **interfaces** and **abstract classes**. We can achieve 100% abstraction using interfaces.

### **How to Achieve Abstraction in Java?**

Java provides two ways to implement abstraction, which are listed below:

* Abstract Classes (Partial Abstraction)
* Interface (100% Abstraction)

## **Java Abstract classes and Java Abstract methods**

1. An abstract class is a class that is declared with an **abstract** keyword.
2. An abstract method is a method that is declared without implementation.
3. An abstract class must have at least one abstract method. And could also have concrete methods
4. Constructors are allowed.
5. There can be a **final method** in abstract class but any abstract method in class(abstract class) can not be declared as final or in simpler terms final method can not be abstract itself as it will yield an error: “Illegal combination of modifiers: abstract and final”
6. A method-defined abstract must always be redefined in the subclass, thus making overriding compulsory or making the subclass itself abstract.
   1. If the **Child class** is unable to provide implementation to all abstract methods of the **Parent class** then we should declare that **Child class as abstract** so that the next level Child class should provide implementation to the remaining abstract method
7. Any class that contains one or more abstract methods must also be declared with an abstract keyword.
8. There can be no object of an abstract class. That is, an abstract class can not be directly instantiated with the *new operator*.
9. An abstract class can have parameterized constructors and the default constructor is always present in an abstract class.