**Abstraction:**

Abstraction is process of hiding the internal details of an object or a class.

Abstraction is process of hiding the internal details and showing functionality is known as abstraction. For example, phone call, We don't know the internal processing about the internal working.

Ways to achieve Abstraction:

1.Abstract class

2.Interface

Hide internal Implementation and just highlight the set of services, is called abstraction

By using abstract classes and interfaces we can implement abstraction.

Example: ATM

ATM GUI screen bank people highlighting the set of services what they offering without highlighting internal implementation.

1). We can achieve security as we are not highlighting our internal implementation.

2). Enhancement will become very easy because without effecting end user we can able to perform any type of changes in our internal system.

3). It provides more flexibility to the end user to use system very easily.

4). It improves maintainability of the application.

**Abstract Class:**

A class which is declared with the abstract keyword is known as an abstract class.

It can have abstract and non-abstract methods (method with the body). It needs to be extended and it’s method implemented. It cannot be instantiated.

Points to remember: 1) An abstract class must be declared with an abstract keyword.

2) It can have abstract and non-abstract methods.

3)It cannot be instantiated.

4) It can have constructors and static methods also.

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

Example of Abstract class that has an abstract method

1. **abstract** **class** Bike{
2. **abstract** **void** run();
3. }
4. **class** Honda4 **extends** Bike{
5. **void** run(){System.out.println("running safely");}
6. **public** **static** **void** main(String args[]){
7. Bike obj = **new** Honda4();
8. obj.run();
9. }
10. }

running safely

**Interface:**

An Interface in java is a blueprint of a class. It has static constants and abstract methods.

The Interfaces in java is mechanism to achieve abstraction. There can be only abstract methods in the java interfaces, not method body.

It is used to achieve abstraction

Java Interface also **represents the IS-A relationship**.

It cannot be instantiated just like the abstract class.

Since Java 8, we can have **default and static methods** in an interface.

Since Java 9, we can have **private methods** in an interface.

Why use Java interface?

There are mainly three reasons to use interface. They are given below.

* It is used to achieve abstraction.
* By interface, we can support the functionality of multiple inheritance.
* It can be used to achieve loose coupling.
* How to declare an interface?
* An interface is declared by using the interface keyword. It provides total abstraction; means all the methods in an interface are declared with the empty body, and all the fields are public, static and final by default. A class that implements an interface must implement all the methods declared in the interface.

**Difference Between Abstract class and interface:**

|  |  |
| --- | --- |
| Abstract class | Interface |
| An Abstract class can have a method body (non-abstract methods). | The Interface has only abstract methods. |
| An Abstract class can have instance variables. | An Interfaces cannot have instance variables. |
| An Abstract class can have Constructor. | An interface cannot have the constructor. |
| An Abstract class can have static methods. | The interface cannot have static methods. |
| You can extend one abstract class. | You can implement multiple interfaces. |
| The abstract class provide implementation of the interface. | The Interface can’t provide implementation of abstract class. |
| The abstract keyword is used to declare an abstract class. | The interface keyword is used to declare an interface |
| An abstract class can extend another java class and implement multiple java interfaces. | An interface can extend another java interface only. |
| An Abstract class can be extended using keyword extends. | An interface class can be implemented using keyword implements. |
| A java abstract class can have class members like private, protected, etc. | Members of a java interface are public by default |
| **Example:** public abstract class Shape{ public abstract void draw(); } | **Example:** public interface Drawable{ void draw(); } |