## Abstract Class

A class that is declared using “abstract” keyword is known as abstract class. It can have abstract methods (methods without body) as well as concrete methods (regular methods with body). A normal class (non-abstract class) cannot have abstract methods.

Abstract class declaration

An abstract class outlines the methods but not necessarily implements all the methods.

Why we need an abstract class?

Let’s say we have a class bird. A bird color changes from bird to bird so we declare the color() method as abstract and we don’t need to implement this in parent class. But birds will fly so this can be implemented as a normal method in parent class.

Parent Class:

**package** FPPackage;

**public** **abstract** **class** Bird {

**public** **abstract** **void** color();

**public** **static** String fly() {

**return** "I fly";

}

}

Child Class:

**package** FPPackage;

**public** **class** Parrot **extends** Bird{

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

// **TODO** Auto-generated method stub

String flystring = *fly*();

Parrot p = **new** Parrot();

p.color();

System.***out***.println(flystring);

}

**public** **void** color() {

System.***out***.println("My color is green");

}

}

Note 1: As we seen in the above example, there are cases when it is difficult or often unnecessary to implement all the methods in parent class. In these cases, we can declare the parent class as abstract, which makes it a special class which is not complete on its own.

A class derived from the abstract class must implement all those methods that are declared as abstract in the parent class.

Note 2: Abstract class cannot be instantiated which means you cannot create the object of it. To use this class, you need to create another class that extends this this class and provides the implementation of abstract methods, then you can use the object of that child class to call non-abstract methods of parent class as well as implemented methods (those that were abstract in parent but implemented in child class).

Note 3: If a child does not implement all the abstract methods of abstract parent class, then the child class must need to be declared abstract as well.

**Why can’t we create the object of an abstract class?**

Because these classes are incomplete, they have abstract methods that have no body so if java allows you to create object of this class then if someone calls the abstract method using that object then What would happen? There would be no actual implementation of the method to invoke.  
Also because an object is concrete. An abstract class is like a template, so you have to extend it and build on it before you can use it.

**Abstract class vs Concrete class**

A class which is not abstract is referred as **Concrete class**.

**Key Points:**

An abstract class has no use until unless it is extended by some other class.

If you declare an abstract method in a class then you must declare the class abstract as well. you can’t have abstract method in a concrete class. It’s vice versa is not always true: If a class is not having any abstract method then also it can be marked as abstract.

It can have non-abstract method (concrete) as well.