

# Abstract Classes in Dart

An **Abstract class** in Dart is defined as those classes which contain one or more than one abstract method (methods without implementation) in them. Whereas, to declare an abstract class we make use of the **abstract** keyword. So, it must be noted that a class declared abstract may or may not include abstract methods but if it includes an abstract method then it must be an abstract class.

Features of Abstract Class:

- A class containing an abstract method must be declared abstract whereas the class declared abstract may or may not have abstract methods i.e. it can have either abstract or concrete methods
- A class can be declared abstract by using **abstract** keyword only.
- A class declared as abstract can't be initialized.
- An abstract class can be extended, but if you inherit an abstract class then you have to make sure that all the abstract methods in it are provided with implementation.

Generally, abstract classes are used to implement the abstract methods in the extended subclasses.

**Syntax:**

```
abstract class class_name {  
  
    // Body of the abstract class  
}
```

Overriding abstract method of an abstract class.

Example

```
// Understanding Abstract class in Dart  
  
// Creating Abstract Class  
abstract class fl {
```

```

        // Creating Abstract Methods
        void say();
        void write();
    }

    class flutterlearning extends fl{
        @override
        void say()
        {
            print("hii");
        }

        @override
        void write()
        {
            print("Flutter & Dart");
        }
    }

    main()
    {
        flutterlearning geek = new flutterlearnig();
        geek.say();
        geek.write();
    }

```

### Output:

```

hii
Flutter & Dart

```