Dart Programming - Interfaces

An **interface** defines the syntax that any entity must adhere to. Interfaces define a set of methods available on an object. Dart does not have a syntax for declaring interfaces. Class declarations are themselves interfaces in Dart.

Classes should use the implements keyword to be able to use an interface. It is mandatory for the implementing class to provide a concrete implementation of all the functions of the implemented interface. In other words, a class must redefine every function in the interface it wishes to implement.

Syntax: Implementing an Interface

```
class identifier implements interface_name
```

Example

In the following program, we are declaring a class **Printer**. The **ConsolePrinter** class implements the implicit interface declaration for the **Printer** class. The **main** function creates an object of the **ConsolePrinter** class using the **new** keyword. This object is used to invoke the function **print_data** defined in the **ConsolePrinter** class.

Live Demo

```
void main() {
   ConsolePrinter cp= new ConsolePrinter();
   cp.print_data();
}
class Printer {
   void print_data() {
      print("__________Printing Data__________");
   }
}
class ConsolePrinter implements Printer {
   void print_data() {
      print("__________Printing to Console__________");
   }
}
```

It should produce the following **output** −

```
__________Printing to Console__________
```

Implementing Multiple Interfaces

A class can implement multiple interfaces. The interfaces are separated by a comma. The **syntax** for the same is given below −

```
class identifier implements interface-1,interface_2,interface_4…….
```

The following **example** shows how you can implement multiple interfaces in Dart −

Live Demo

```
void main() {
   Calculator c = new Calculator();
   print("The gross total : ${c.ret_tot()}");
   print("Discount :${c.ret_dis()}");
}
class Calculate_Total {
   int ret_tot() {}
}
class Calculate_Discount {
   int ret_dis() {}
}
class Calculator  implements Calculate_Total,Calculate_Discount {
   int ret_tot() {
      return 1000;
   }
   int ret_dis() {
      return 50;
   }
}
```

It should produce the following **output** −

```
The gross total: 1000
Discount:50
```