Can a class implement more than one interface? B10. Can a class extend more than one class in Dart?

Ans:

Yes, in Dart, a class can implement more than one interface. Dart supports multiple interface inheritance, which allows a class to implement multiple interfaces, each specifying a set of method signatures that the implementing class must provide implementations for. This is a form of multiple inheritance without the complexities associated with multiple class inheritance.

Ex:

abstract class Swimmer {

void swim();

}

abstract class Flyer {

void fly ();

}

class Bird implements Swimmer, Flyer {

void swim () {

print('Bird is swimming.');

}

void fly () {

print('Bird is flying.');

}

}

In this example, the 'Bird’ class implements both the 'Swimmer' and 'Flyer" interfaces. It

provides implementations for the swim () and fly () methods specified by these interfaces.

B10 :

No, a class cannot extend more than one class in Dart. This is known as the single inheritance principle.

There are a few reasons why Dart does not support multiple inheritance:

• It can lead to ambiguity and confusion. For example, if a class extends two classes that both have the same method, which method should be called when the method is invoked?

• It can make code difficult to maintain and update. If we need to change the behavior of a class that extends multiple classes, we need to make sure that the changes are compatible with all of the classes that the class extends.

• It can lead to performance problems. When a class extends multiple classes, the compiler needs to generate more code to support the multiple inheritance.