



Can you explain to me how it's working 'Interfaces' in easy and detailed samples in c++ code.



Sure, I'd be happy to explain interfaces in C++!

In C++, an interface is a class that contains only pure virtual functions, with no implementation. A pure virtual function is a function declared with the `virtual` keyword and set to `0`. It's a way to specify a function in a base class that must be implemented in any derived classes.

Here's an example of how to declare an interface in C++:

C++

Copy code

```
class Shape {  
public:  
    virtual double area() const = 0; // pure virtual function  
    virtual double perimeter() const = 0; // pure virtual function  
};
```

In this example, `Shape` is an interface with two pure virtual functions: `area()` and `perimeter()`. Any class that inherits from `Shape` must implement these functions, or else it will also be an abstract class.

Let's say we want to create a `Rectangle` class that inherits from `Shape`. Here's an example implementation:

C++

Copy code

```
class Rectangle : public Shape {  
private:  
    double length_;  
    double width ;
```

