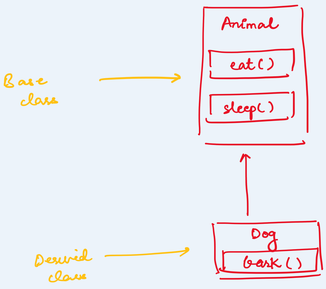
Inheritance

It allows us to create a new class (derived) from an existing class (base)

```
class Animal {
    // eat()
    // sleep()
};
class Dog : public/private/protected Animal {
    // bark()
};
```



* Inheritance is an **is a relationship**

We can only use Inheritance only if an is-a relationship is present b/w two classes.

→ A car is a vehicle

→ A surgeon is a doctor.

C++ protected members

protected members are inaccessible outside of the class.
However, they can be accessed by derived classes & friend functions.

Access modes

1. public → members of base class are inherited by derived class just as they are.
2. private → all members of base class becomes private members in derived class
3. protected → public members of base class become protected members in derived class.

C++ function overriding

If same function is defined in both the derived class and the base class, & if we call this function, using the object of derived class, the function of derived class is executed.

```
class Base {
    public:
        void print() {
        }
};
class Derived : public Base {
    public:
        void print() {
        }
};
int main() {
    Derived derived1;
    derived1.print();
    Base b1;
    b1.print();
}
```