Abstract keyword

Lets assume as class car it ~~has~~ tells that it has following features of that it but the implementation of that feature is not there.

So, if we create any Object and call that feature method, nothing would happen.

So no one will interested in creating that object.

So anyone need this class?

Yeah, ?

when someone wants to design car and they know that it will drive, but how it will drive, they don't know,

So, they just declare that method so that whenever some one is ~~to~~ implementing Car, they will define how car drive.

But when we just declare it will throw error?

→ So, we will make that method abstract

Still there is ~~an~~ error on drive as well as on class.

→ ~~abt~~ Abstract method only defined ~~is a~~ by an abstract class

The class which is extending must define abstract method.

↑
So, class should be abstract

i.e.

```
abstract class Car {
    public abstract void drive(); // declaration
    // it can have non abstract method

class WagonR extends Car {
    public void drive() {
        sout("driving");
    }
}
```

```
~~Public~~ Class Demo {
    main() {

    X Car obj = new Car(); X
```

we can make car as reference

```
    Car obj = new WagonR();
    obj.drive();
```

Note

~~So~~ It is necessary to define the abstract method in ~~[illegible]~~ child class.?

Yeah, but we can also not define all the abstract method in child class, ~~concrete class~~

Only if child class is also abstract.

So, we ~~also~~ also need to ~~[illegible]~~ extend that abstracted child class to implement the abstract method

Reason:- we can't create object of that abstract child class.