#Abstract classes In JAVA

Types of classes → (A) abstract class
(B) Concrete class → normally created class

Ex:-

```
class hello{
}
```

= this above is a example of ~~abstract~~ concrete class =

```
abstract class hello2{
}
```

= this above is example of = abstract class

→ you can't able to create objects for concrete classes where you do it for concrete class but creation of reference is allowed

hello2 a; it is allowed

hello2 a = new hello2(); is not allowed.

→ for concrete both are allowed.

you have to learn things

→ (A) abstract method (B) Abstract class

abstract method is a method without having a body and has keyword abstract before return type

Ex: abstract void helloworld();

→ helloworld function don't have body

→ Abstract methods can be ~~defined~~ called as undefined methods

→ If something in class is not abstract that is any data member or member function/method is abstract whole class ~~is abstract~~ you have to make it abstract else it will give error

→ A abstract have 0 or more abstract methods

→ so when we need abstract classes

abstract are meant only for inheritance

→ So you can declare a ~~fun~~ class as abstract but if you want to declare a concrete class having abstract class as it's parent you have to do method overriding for all its abstract method.

so

```
abstract class parent{
    ~~to~~ abstract void hell();
}
```

option 1:

```
class child extend parent {
    public void hell(){
        code
    }
}
```

option 2

```
abstract
  ~~class child extend~~
abstract class
child extends parent{
  abstract void hell();
}
```

now child is concrete class as it ~~de~~ declared abstract function with method overidn

Else you have to create a abstract class.

→ overiding is not compulsion here

→ Here for concrete child class you can create objects.

also you can reference it parent abstract class with that child (Dynamic method dispatch)

~~to~~

→ Dynamic method dispatch can be used for abstract class

```
Parent a = new child();
```

where parent is abstract & child is concrete class

→ Also child can access functions inside parent normally as we do for concrete parents

→ If you're working under some project & you have some goals & so we can create abstract class & store the function goals as function & when we doing implementation just declare fun

→ You can declare functions inside abstract class & write code in child body

→ for each child you have to do complete coding for all the ~~codes~~ abstract function even if one abstract function is not overrided it will give error.

→ If you want some mandatory function in class say for polygon area & perimeter so you can do it with abstract classes

→ see code & notes in abstract class folder

→ An abstract class also can have a constructor

⊙ Rules to abstract class

abstract ~~will never become~~ static or final functions

class can't have

→ abstract class only meant for inheritance