Abstract Class
Abstract Methods *

Unit-2

- A class w/ atleast 1 abstract method is called <u>ab. class</u>.
- Abstract methods are <u>undefined</u> methods. They r declared w/ abstract keyword.

How to create abs. class

- Create <u>atleast 1 abs.</u> method in the class
- It can have other concrete (defined) fun.
- It must be declared w/ <u>abs. keyword</u>.

eg
```
abstract class myclass {
        abstract public void get();
        //abstract method - it is undefined &
            abs. KW is used.
}
```

what is the use of abs: class

- An abs. class is meant to be <u>inherited</u>
- An abs. class provides <u>abstraction</u>
- When any class inherits an abs. class, then it <u>must define</u> the abst. fun. The fun param & name must be same.

Rules

What is allowed in abs. class:

- It can have both abstract & concrete methods.
  - However there must be atleast 1 abs. method
- Constructor
- Static meth - both abs & concrete can be static
- final " - only concrete meth " " final
- abs class is meant to be inherited

What is not allowed:

- There must be atleast 1 abs method in abs class
- *It can't be intantiated - means object of abs. class can't be created bcoz it has undefined fun.
- Ref. of abs. Class can be created.
- If derived class doesn't implement the abs. method, then it must itself be declared as abstract class.

Eg:- Show the use of abs. class.
It ~~can~~ should contain constructor, abs, concrete, final, static methods.

```
abstract class base {

    //abs fun
    abstract public void get();

    //concrete fun
    final public static void fun(){
            Sopln("fun is called");

    }

    //constructor
    base(){
            Sopln("Constructor called");
    }
} // base class (abs)
class box extends base {
        protected int l,w,h;
        public void set(int x, int y, int z){
             l=x; w=y; h=z;

        }

        // since box extends base, it must
        define/override get() with same name & param.
        public void get(){
              Sopln(l); Sopln(w); Sopln(h);

        }
} // box ends.
```

```
class demo{
        p s v m (--) {
              //obj of base can't be created
              box obj = new box(); //Constructor of base
                                     (abs) will be called
              obj.set(1,2,3);        bcoz of Constr chaining
              obj.get();

              //Static fun of base (abs) can be called
              ~~box~~ base.fun();
        }
}
```

- - - - - - - - - - - - - - - -