Day19

ABSTRACTION

- Hiding the implementation details from user
- only functionality will be provided to user

To Achieve abstraction :-

1. Abstract class (0–100%) (NOT 100%)
2. Interface (100%)

Advantage

→ increase security
→ reduce complexity

1. Abstract class:

- use 'abstract' keyword to declare Abs class/meth
- Abstract method (meth with body)
- non-abstract method (metho without body)
- cannot create abstract class object
- Implement abstract method using Inheritance (so in child class we can create obj)

eg: public void display();

abstract void display();

eg 2:
```
abstract class google {

    abstract void search();

    public void maps();
}
```

```
sysout("google search");
    }
}
class SearchAll extends google{
//{(Add unimplemented method).
}
class SearchImage extends google{
//(Add unimplemented method)
}
class searchVideo extends google{
    //(Add unimplemented methods)
}.
PSVM.
// (Object creation for child class) & call it
```

why it is not 100%?

- non abstract method we have given implementation prams

(2) INTERFACES:

- 10 modules (designed for multiplication 5 modules)
- Remaining 5 modules we can create Interface
- without implement we will use method.

```
interface Test
{
    void display();
    void m1();
}
```

- we have .interface interfacename
- All method in interface are abstract

- by default it becomes public
- All variables in interface are static & final
- cannot create objects
- we can create using implements keyword
- Multiple & hybrid inheritance can achieved through interface

Static keyword:

- only allocating once, no object creation. (obj independent)
- we can access using class name.
- memory Management
- values can be changed.

```
static String dept = "eng";
```

eg: class schooldetails

```
{
static string departname = "English";
}
```

```
PSVM
{
sysout (schooldetails.departnm);
sysout (schooldetails.depart = "real ayaku");
sysout (schooldetails.departnm);
}
```

Final keyword:

- used to store constant value. (no changing)
- obj dependent (obj creation) (each)
- no overriding (method)
- can't extend (class)

eg: class SD {

final String schoolname = "Luminu";

}

PSVM {

SD S = new SD;

sysout (S.schoolname);

// sysout (S.schoolname = "abc"); → error

}