

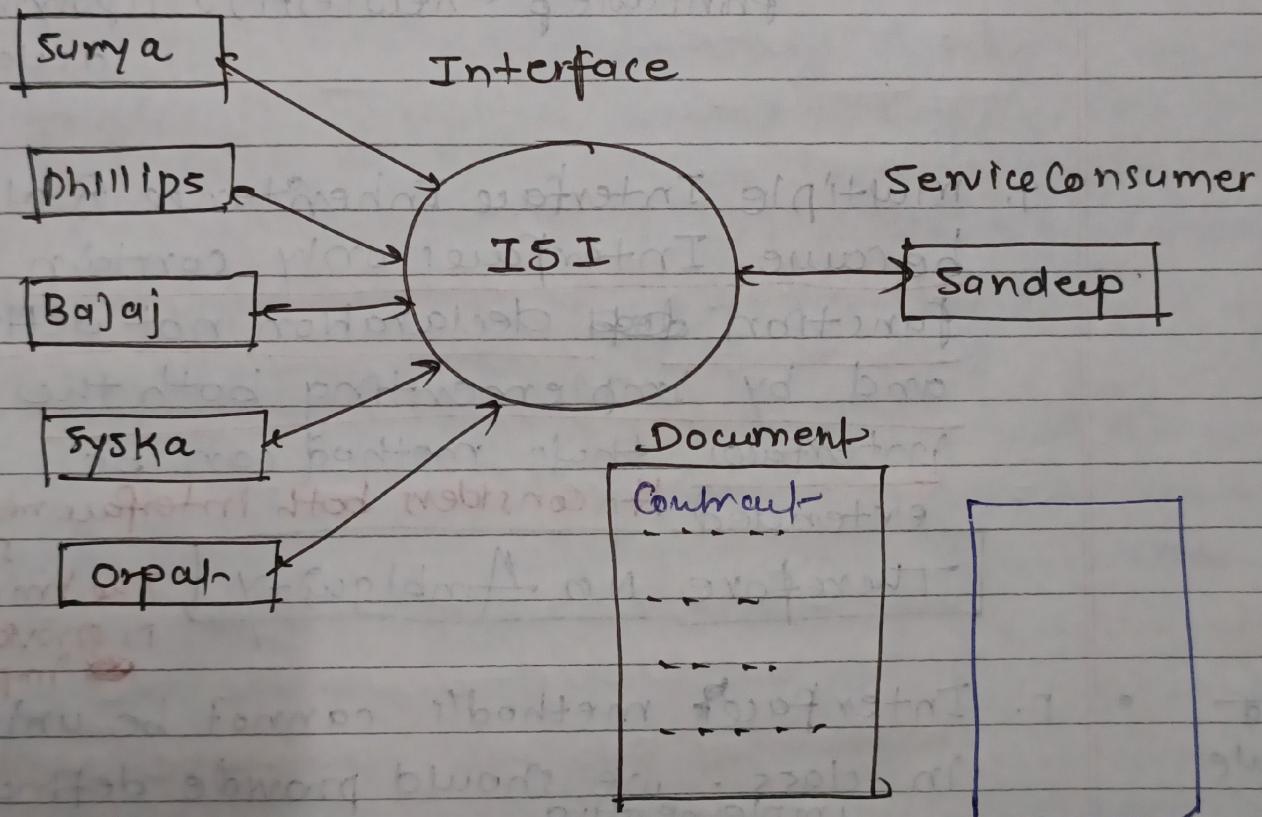
INTERFACE

Page No.

Date

Fragile Base class Problem -

- If changes are done in super-class methods (signatures) then it is necessary to modify and recompile all its subclasses
This is called as "Fragile Base class problem"
- This can be overcome by using Interfaces



class - class

(extends)

Interface - Interface

(extends)

Class - Interface

(implements)

(It is recommended to not make changes in original interface)

If you want to then create another interface which will extend older interface

// Imp ***

1. Fields in Interface are by default

public static final

2. methods in Interface are by default

public abstract

3. we cannot make object of Interface
but we can make reference of Interface

Printable p = new Test(); // polymorphic declaration

4. multiple Interface Inheritance is allowed
because Interfaces only contain
function ~~def~~ declaration not definition
and by implementing both the
interface that method can be

~~extended~~ It considers both Interface methods as

Therefore No Ambiguity

The implementation
is provided in
~~Implementation~~

Rule

• r. Interface's method's cannot be unimplemented
in class. we should provide definition
by implementing the class from interface

(//demo 13)

List 1) Interface ArrayList Class

list = new ArrayList(); ✓

ArrayList list = new ArrayList(); ✗

Page No.

Date



no need of interface
then why above
syntax is recommended.

Abstract method - (logically 100% Incomplete)

- Abstract- method doesn't have definition
- If class contains one or more abstract methods, then class must be declared as abstract. Otherwise compile error.
- The superclass abstract- methods must be overridden in subclass, otherwise subclass must be marked abstract.
- The abstract- methods are forced to be implemented in subclass. It ensures that subclass will ...
- The abstract- method cannot be private, final, static
- example- abstract methods declared in Number class are abstract

Abstract class - (logically incomplete)

- If implementation of a class is logically incomplete, the the class should be Abstract
- If class contains one or more abstract methods, then class must be declared abstract
- An abstract class can have zero or more abstract methods
- Abstract class object cannot be created, but reference can be created.
- Abstract class can have fields, methods and constructor
- example

java.lang.Number
java.lang.Enum

Ques. what is difference between Abstract Class & Interface?

Page No.		
Date		

VVV Imp

(demo 14)

How

- ⊕ If you add method in Interface, then it should be override.

① Class A contains 3 methods F1, F2, F3

F3 is abstract

∴ abstract method इसी ने class A का abstract बनाया।

F1 method is logically complete so final

• F2 method is logically partially complete

② class B inherited from A -

Now class A has F3 method abstract so

it is compulsory to override the F3 method in subclass, otherwise error.

'F2' was partially complete, so in class B it becomes 100% complete
∴ final F2 in class B

③ class C extends B

F3 becomes logically 100% complete

so final and everything in class C is final
∴ class C is final.

Abstract Class

① Object of Abstract class
cannot be created

② Abstract class has
constructors

③ Zero or more abstract
methods

④ access modifiers can
be public, private, protected

⑤ Is a Relationship

Interface

① object of Interface
cannot be created. but
reference can be created.

② Interface do not have
constructors

③ methods are by default
abstract

④ All members are by
default public.

⑤ Can do relationship p-