Pg1

Thursday, July 2, 2020 1:09 PM

Abstraction

It is a process of exposing, the most essential features to the service user, and hiding their implementation details, is known as Abstraction.

In java, we can achieve abstraction with the help of abstract class or interface.

With respect abstraction we can generally classify methods into :

1. Concrete method
2. Abstract method

1. **Concrete Method :**

A method which has implementation (method definition) is known as concrete method.

2. **Abstract Methods :**

A method which does not have implementation (no method definition) is known as abstract method.

How to create abstract method ?

syntax:

```
abstract   return_type   method_name( formal_arguments, … ) ;
```

Rules :

1. Method should not have body.

2. it should be prefixed with modifier **abstract.**
3. the method declaration statement must end with a semicolon.

In java, we can achieve abstraction with the help of :

1. Abstract Class
2. Interface

abstract class :

A class prefixed with abstract modifier is known as abstract class.

```
class Demo
{

}
```

concrete class

```
abstract class Demo
{

}
```

abstract class

Note:

We cannot instantiate an abstract class. (We cannot create an object for abstract class)

When to make a class abstract ?

1. if a class has at least one abstract method, it is mandatory to make the class abstract. If not we get a compile time Error.
2. We should make the class abstract, if the class inherits the abstract method and it is not overridden.

Note :

1. abstract class can have static and not static variables.
2. abstract class can have static methods
3. abstract class can have static and non-static blocks
4. abstract class will have constructor.
5. abstract class can have abstract and concrete non-static methods
6. **we cannot create an object for abstract class.**

<u>How to use a non-static member of abstract class ?</u>

Note :

We can use the non-static member of the abstract class, by creating an object for concrete implementing sub-class.

Example refer, **app24/abstr2/src/Demo7.java**

<u>Implementing class / Concrete implementing class :</u>

the sub-class which provides implementation to the inherited abstract methods is known as implementing class

<u>Concrete Class :</u>

The class which does not have any abstract methods either declared or inherited, and not prefixed with abstract modifier is known as concrete class.

Note :

We can create objects only for concrete class.

Assignment :

1. What is an abstract method ?
2. What is a concrete method ?

3. What is abstract class ?
4. What is concrete class ?
5. When should we make a class abstract ?
6. What is the difference between abstract and concrete method ?
7. What is the difference between abstract and concrete class ?

<u>Can we make a static method abstract ?</u>

We, cannot make a static method abstract. (Abstraction concept is applied only for instance methods/non-static methods)

Reason :   static methods cannot be overridden, hence we cannot provide implementation from the subclass if it is made abstract.
Therefore, static methods cannot be made abstract.

Pg2

Monday, July 6, 2020 1:04 PM

Interface

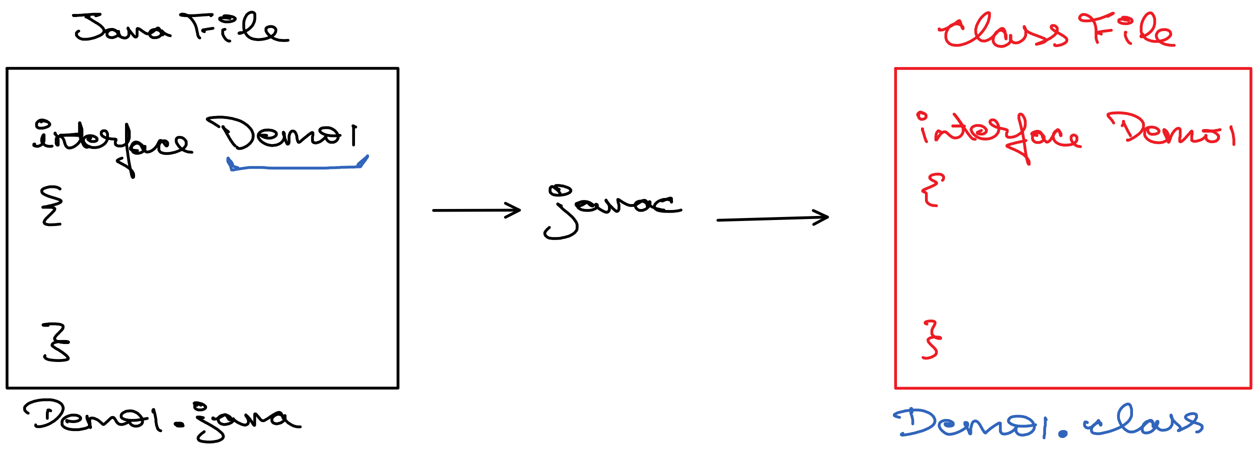
interface is a component of java, which helps programmers to create a non-primitive datatype.

Note :

1. interface is a keyword.

Syntax:

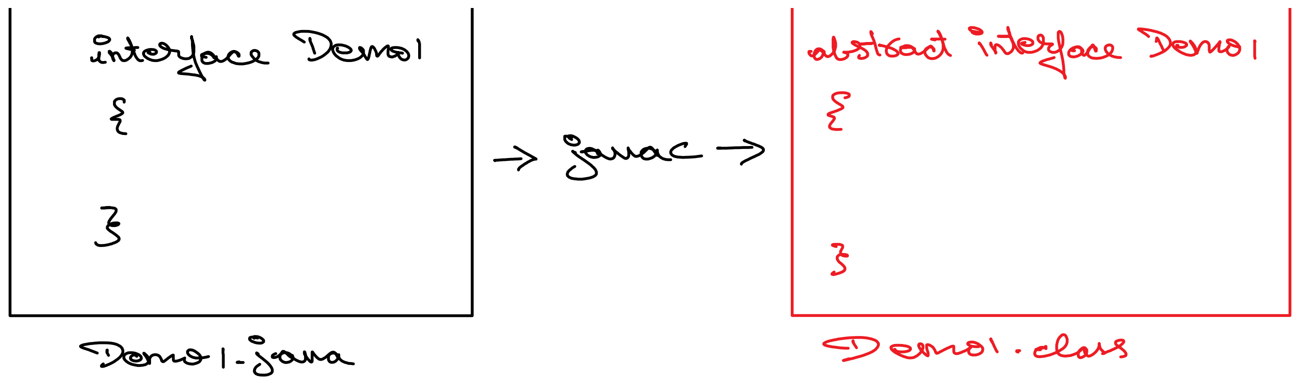
```
interface  Interface_name
{
        // declaration statements ;
}
```



Note :

1. When we compile an interface, a class file is generated, whose name is same as the name of the interface.
2. We can use interface name as the non-primitive data type.
3. interface is by default abstract. (at the time of compilation, the compiler will prefix the modifier abstract to an interface)

interface Demo1
abstract interface Demo1



4. We cannot instantiate an interface. (We cannot create an object for the interface)
5. We can have only package scope or public scope for an interface, we cannot make an interface private or protected.
   for examples refer, app24/abstr3/src/Demo4.java… Demo7.java

==
What is the purpose of Interface ?

1. It is used to achieve abstraction
2. interface helps programmers to achieve 100% abstraction
   (non-static concrete methods are not allowed in interface, in interface we can only have non-static abstract methods also known as interface methods.)
3. To achieve multiple inheritance in java it is possible only with the help of interface.

==

What is allowed in interface up to JDK 1.7 ?

Note : all members of interface are by default have **public scope**.

1. Only a public static final variable is allowed, no other variables are allowed.. examples refer :

   app24/abstr3/src/Demo8.java …Demo10.java

2. In interface we can have only public abstract non-static methods

Note :

Note :

Interface has no constructors

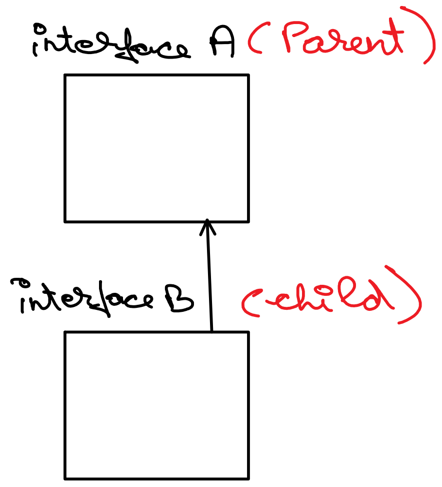
Pg3

Tuesday, July 7, 2020 1:06 PM

Inheritance with respect to interface :

1. interface can inherit any number of interfaces.

case1 :

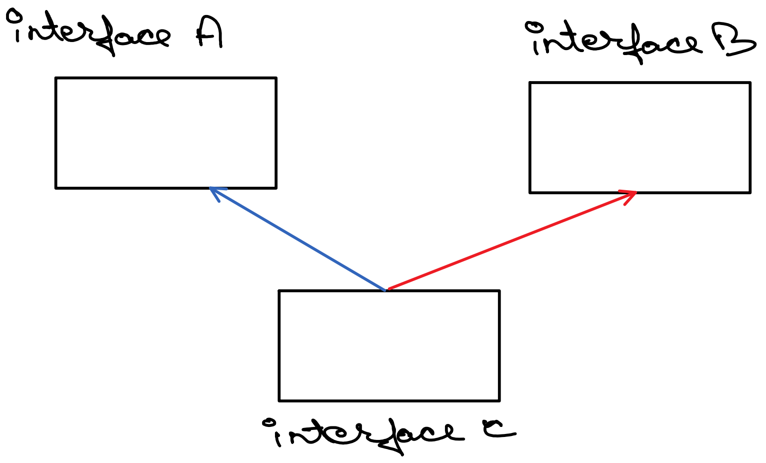


```
interface A
{

}

interface B extends A
{

}
```

an interface can inherit another interface, with the help of **extends** keyword.

Note : We can achieve multiple inheritance with the help of interface.

case2 :



```
interfac A
{
```

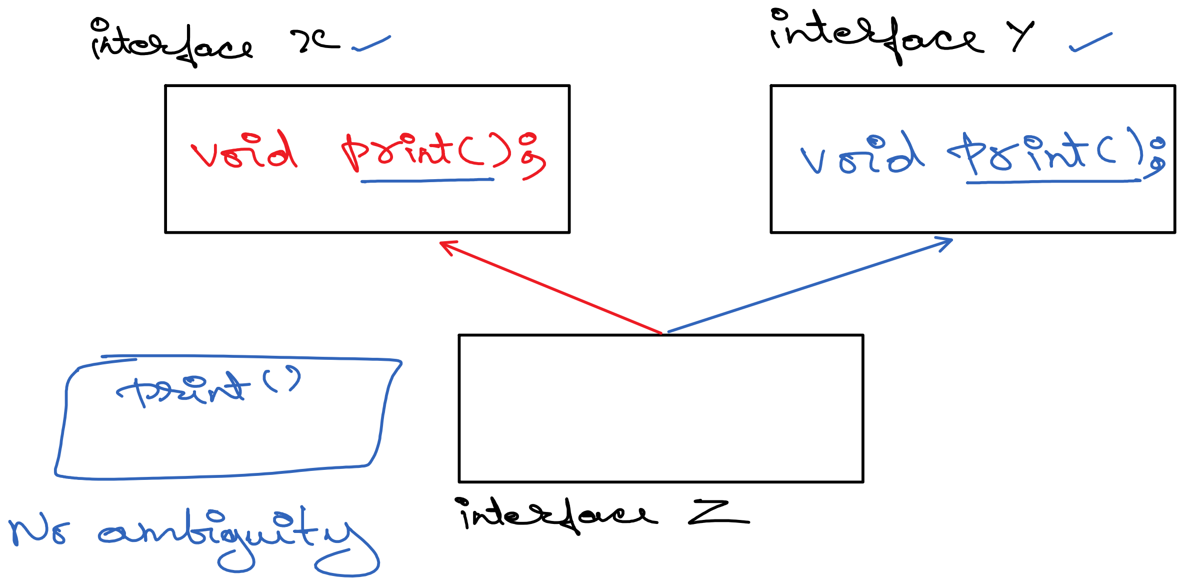
```
interface C extends A, B, ...
```

```
interfac A                interface C extends A , B , ...
{

}

interface B
{

}
```

Note : In interface we don't get diamond problem for multiple inheritance.

Reasons :

1. No constructor in interface, therefore no ambiguity due to super() statement.
2. the interface abstract methods, do not have implementation, hence no ambiguity

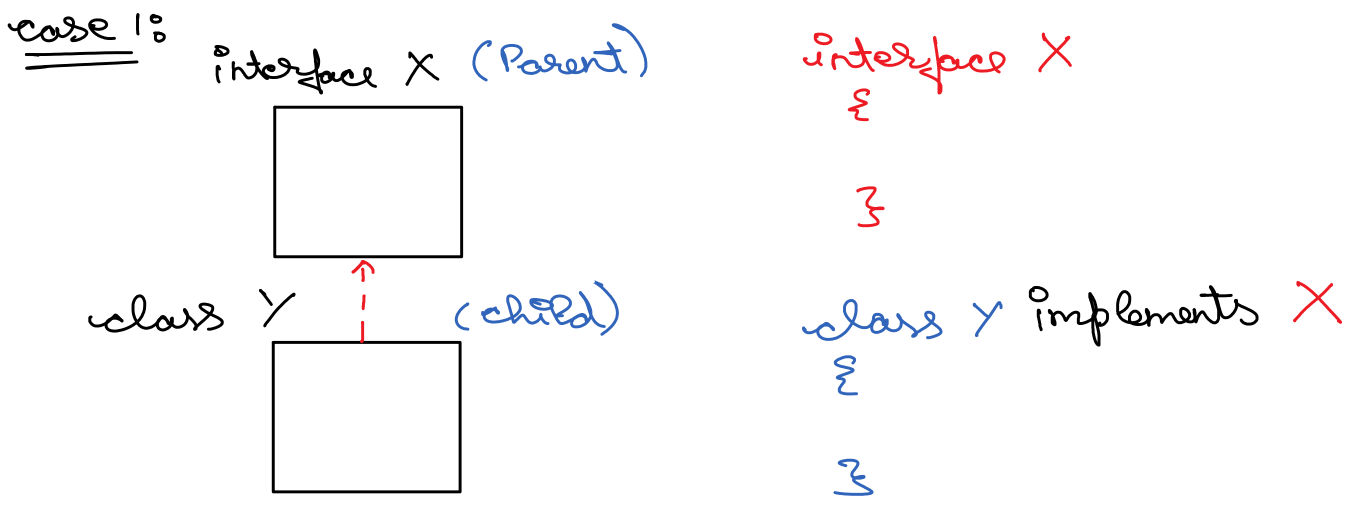


Pg4

Tuesday, July 7, 2020 1:46 PM

2. The inheritance between interface and class :

2.1 A class can inherit an interface :



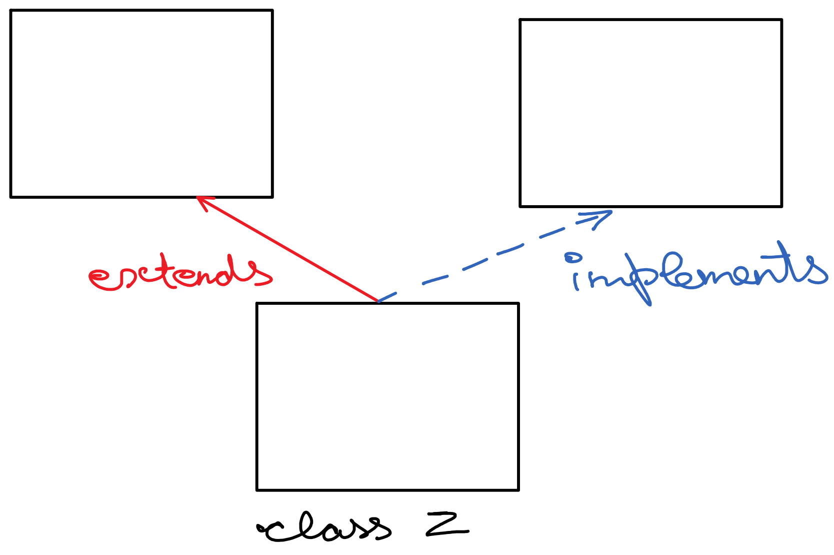
Note :

a class can inherit any number of interfaces with the help of implements keyword

```
class class_name  implements  interfac1 , interface2, …
{

}
```

2.2 A class can inherit a class as well as an interface.

class X
interface Y



Rule :

1. always extends should be written first and then implements, else we get CTE.

```
class X                         interface Y
{                               {

}                               }
```

case1:

```
class Z extends X implements Y
{
}
```

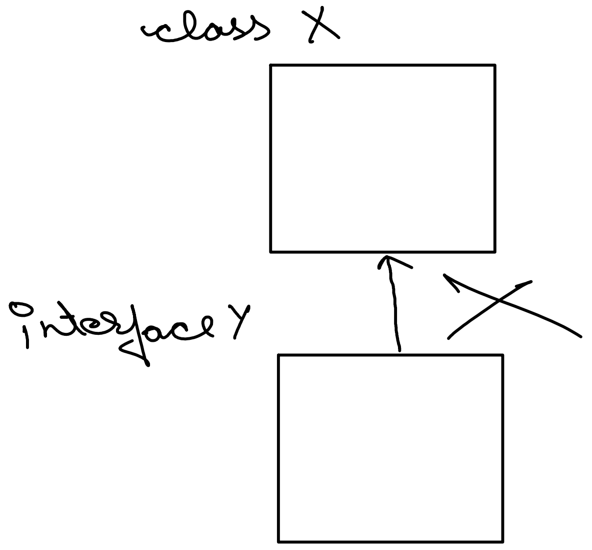
✓ successful

case2:

```
class Z implements Y extends X
{

}
```

✗ CTE

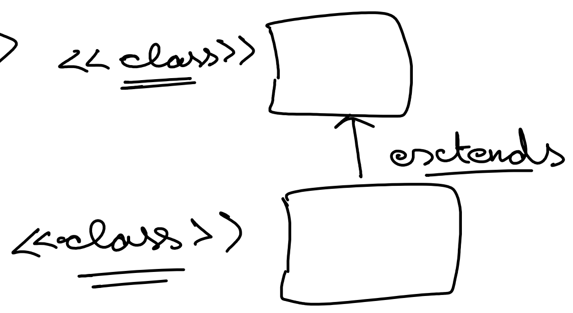
2.3 Can an interface inherit a class ?

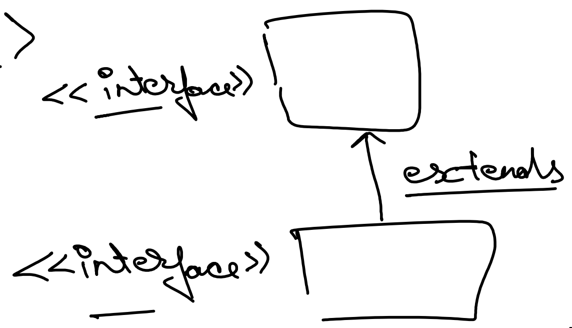


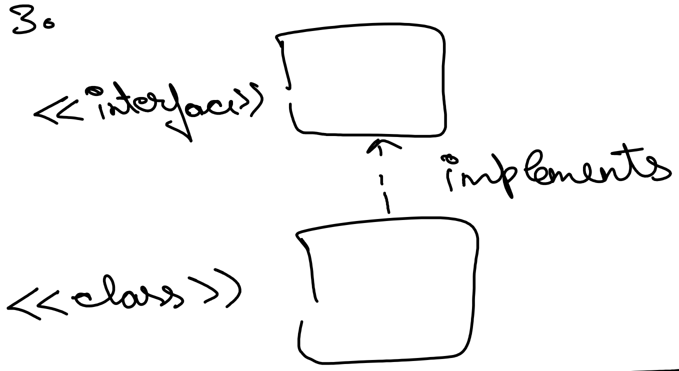
? X No not possible

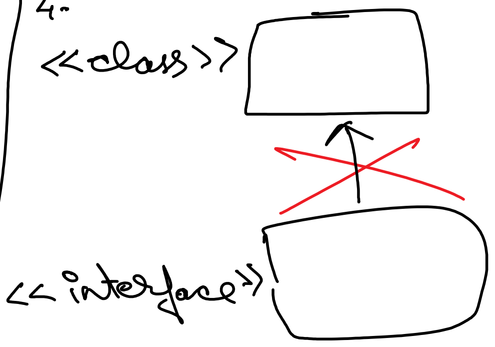
Pg5

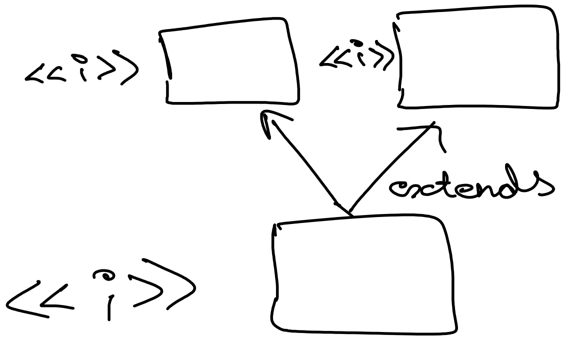
Tuesday, July 7, 2020 2:06 PM

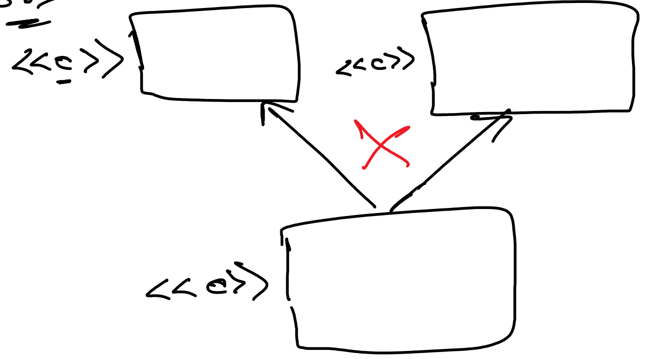
1.)


2.)


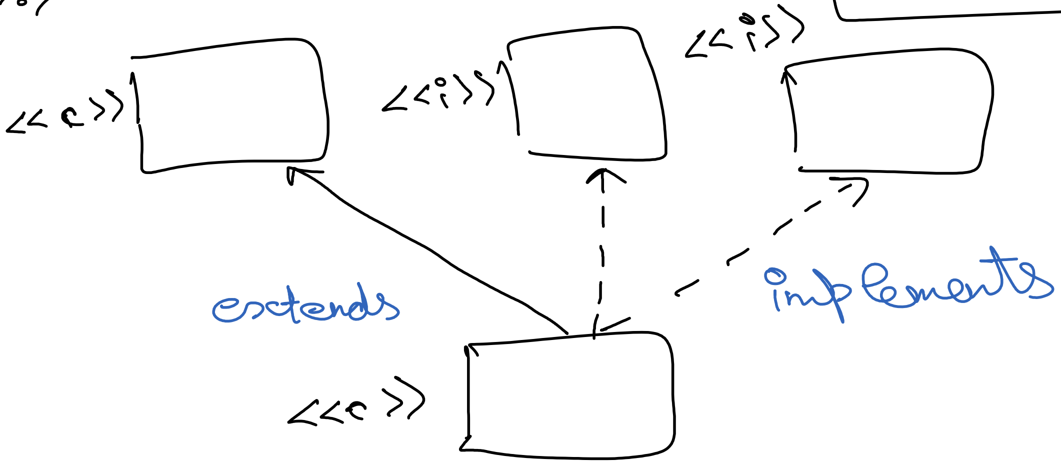
3.


4.


5.)


6.)


A class can inherit only one class

7.)


Pg5

Tuesday, July 7, 2020 2:14 PM

10)

```
interface Student
{
        void bunk();
}
```

Pg6

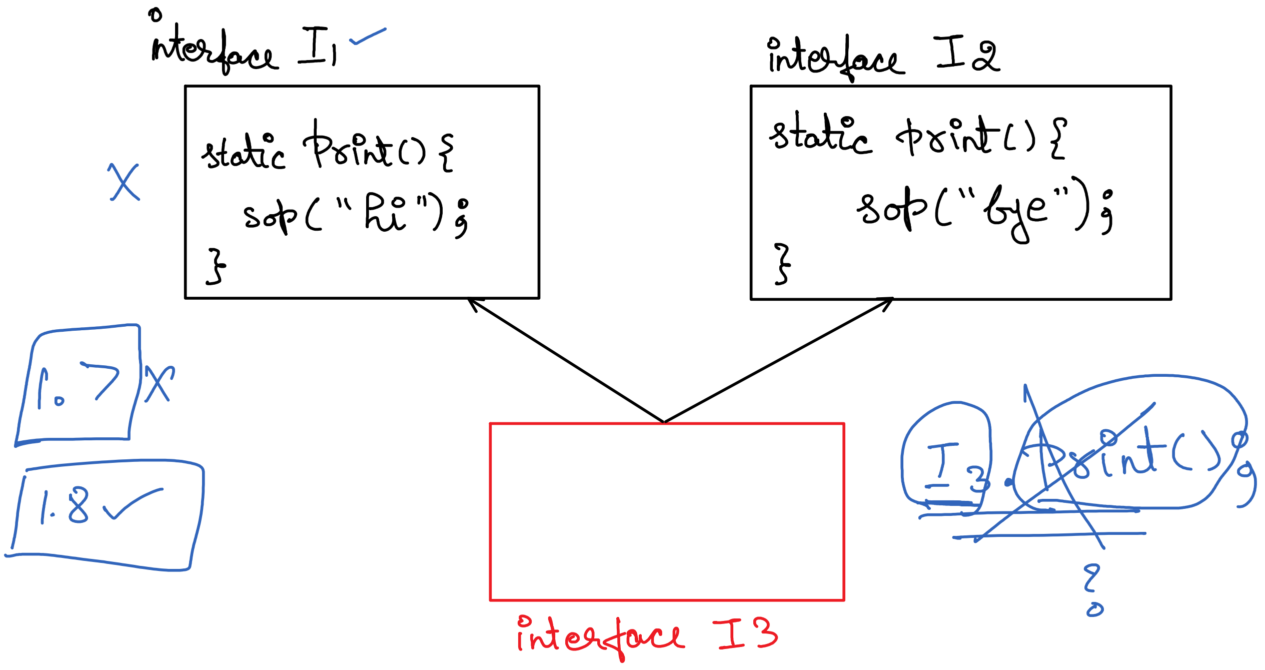
Wednesday, July 8, 2020 1:39 PM

In JDK 8 :

1. static methods were introduced into the interface.

Note :

1. We can have static methods defined in interface from JDK 8 onwards.
2. all static methods in interface are by default public.



Note :

1. Static methods of interface is not inherited.
2. There we will not have diamond problem.

==

Task1:

1. What is Abstraction ? how do we achieve it?
2. What is the difference between abstract class and concrete class ?
3. What is the difference between abstract class and interface ?

Assignment Questions :

1. What is Abstraction ?
2. How do we achieve abstraction in java ?
3. what is an abstract method ?
4. what is the difference between abstract and concrete method ?
5. What is an abstract class ?
6. Is it compulsory to make all the classes abstract ? explain with example.
7. What is the difference between abstract class and concrete class ?
8. What is interface ?
9. What is the difference between interface and abstract class ? (Why do you want an interface when we have abstract class ?)
10. What is the difference between extends and implements ?
11. Explain multiple inheritance with an example ? and explain the special characteristic of multiple inheritance.
12. What is the difference between encapsulation and abstraction ?