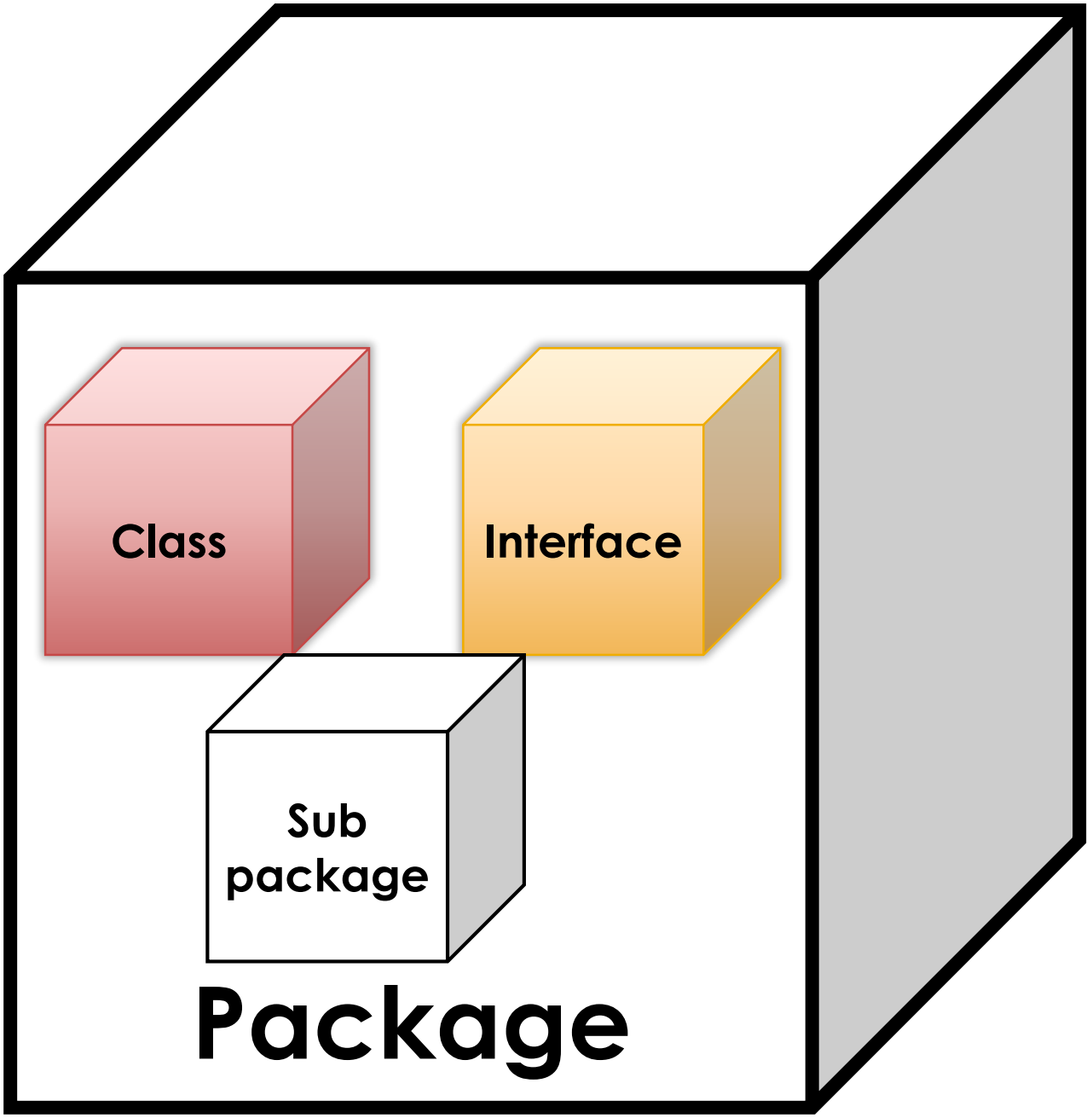
Object Oriented Programming

Interface

Course Instructor: Sadaf Anwar

Interface

- What are interface?
- Why do we need it?
- Syntax?
- Implementation


Class
Interface
Sub
package
Package

What is interface in OOP?

1. Abstract class provide partial abstraction
2. It is like a class
3. the variables declared in an interface are public, static & final by default
4. Writing an interface is similar to writing a class. But a class describes the attributes and behaviors of an object. And an interface contains behaviors that a class implements

Syntax

```
interface <interface_name>{

    // declare methods that abstract
    // by default.
}
```

Example

```
interface example_interafce
{
   public void method1();
   public void method2();
}
```

Interface must be implemented

The class that implements interface must implement all the methods of that interface.

Implementation of Interface

```
1. interface my_animal {
2.     public void eat();
3.     public void travel();
4. }

5. public class demo implements my_animal {
6. public void eat() {
7.     System.out.println("Mammal eats");
8. }
```

```
9.   public void travel() {
10.              System.out.println("Mammal travels");
11. }
12. public int noOfLegs() {
13.              return 0;
14. }
15. public static void main(String args[]) {
16. demo m = new demo();
17. m.eat();
18.  m.travel();
19. }
20. }
```

A class can Implement Multiple Interfaces

```
1. Interface A{
2. Public method_A();
3. Public method_B();
4. }
5. Interface B{
6. Public method_C();
7. }
```

```
1.  public class MyInterface implements A, B{


2.      public void method_A() {
3.          System.out.println("Hello");
4.      }
5.      public void method_B() {
6.          System.out.println("Goodbye");
7.      }
8.     public void method_C() {
9.          System.out.println("Goodbye");
10.     }
11. }
```

Interface extends interface

```
1. public interface A {
2. void meth1();
3. void meth2();
4. }
5. public interface B extends A{
6. void meth3();
7. }
```

```
public class C_A_B implements B{
    public void meth1()
    {
        System.out.println("Implement meth1().");
    }
    public void meth2()
    {
        System.out.println("Implement meth2().");
    }
    public void meth3()
    {
        System.out.println("Implement meth3().");
    }
}
```

```
public class C_ab_main {

    public static void main(String args[])
    {

        C_A_B cab=new C_A_B();
        cab.meth1();
        cab.meth2();
        cab.meth3();
    }
}
```

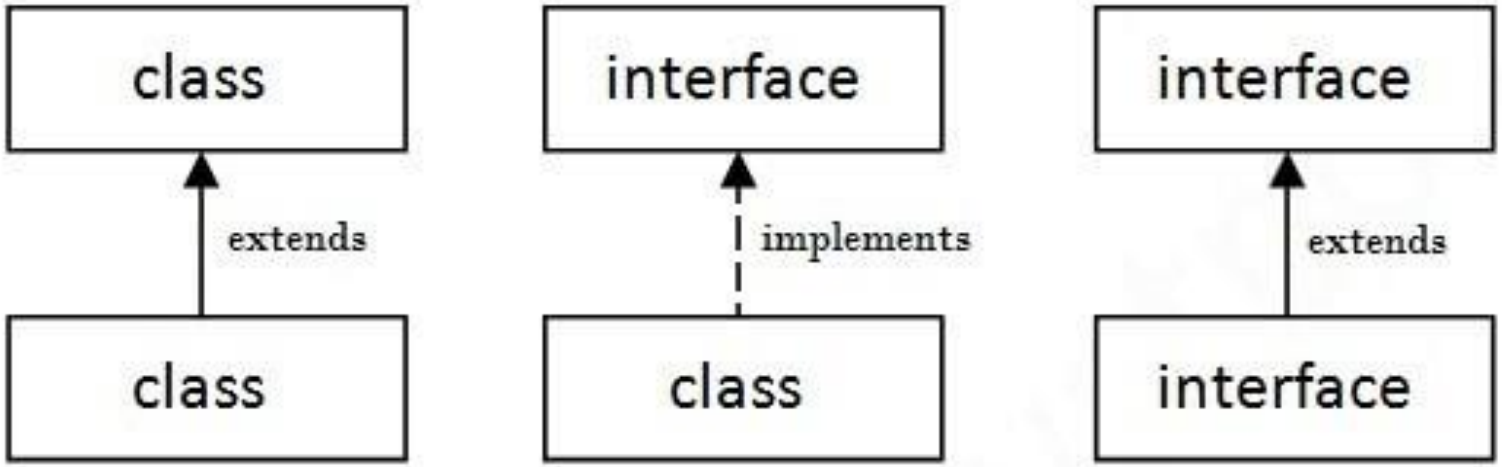
Why do we need interface?

1. When we want **total abstraction.**
2. Since java does not support **multiple inheritance** in case of class, but by using interface it can achieve multiple inheritance.
3. It is also used to achieve **loose coupling**.

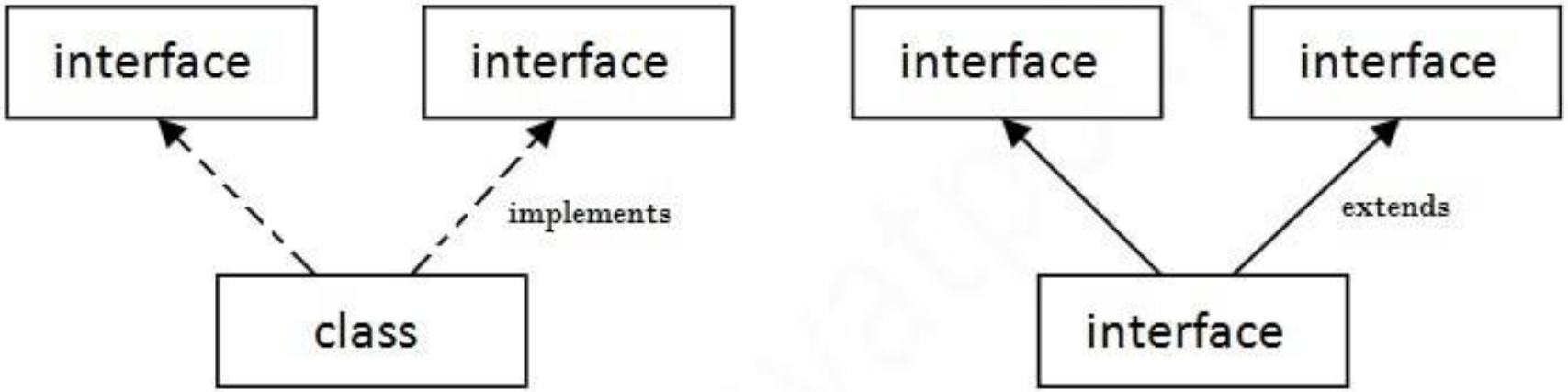
Interfaces are used to implement abstraction. So the question arises why use interfaces when we have abstract classes?

The reason is, abstract classes may contain non-final variables, whereas variables in interface are final, public and static.

So we can say


class
extends
class
interface
implements
class
interface
extends
interface

Multiple inheritance can be achieved by interface in java



Multiple Inheritance in Java

```
1.interface X{
2.public void myMethod1();
3.}
4.interface Y{
5.public void myMethod1();
6.}
7.class Demo implements X, Y{
8.public void myMethod1(){
9.System.out.println(" Multiple
 inheritance example using
 interfaces");
10.}}
```

Your task

- Difference between loose coupled and tightly coupled with code example.

```
class Course {
    Topic t = new Topic();
    public void Reading()
    {
        t.understanding_concept();
    }
}
class Topic {
    public void understanding_concept()
    {
        System.out.println("Tight coupling concept");
    }
}
```

Summary

Questions