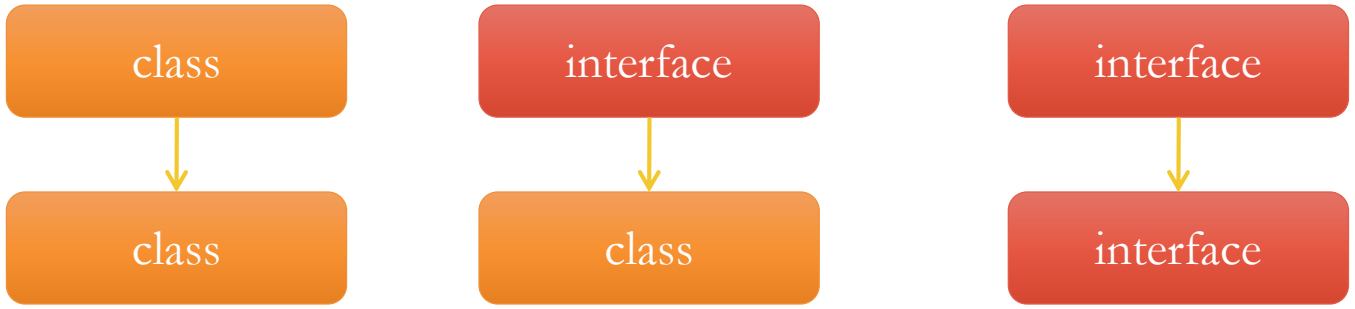
Java

Agenda

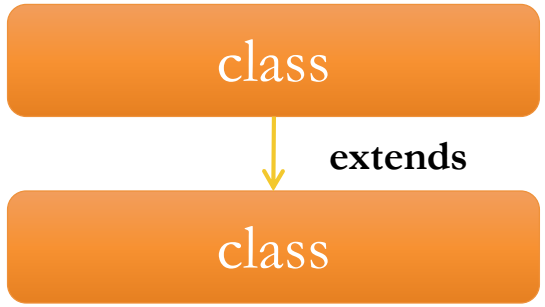
- Java Interfaces
- Java Packages
- Access Modifier's

Java Interface

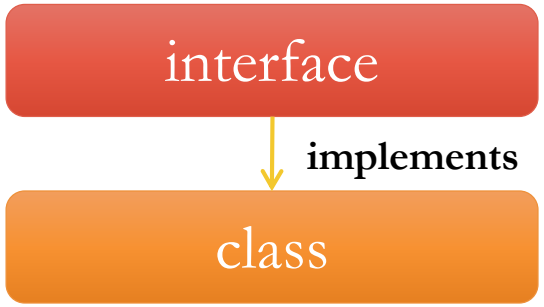
- An **interface in java** is a blueprint of a class.
- Interface contains **final and static variables**.
- Interface contains **abstract methods**.
- An **abstract method** is a method contains definition but not body.
- Methods in interface are public by default.
- Interface supports the functionality of multiple inheritance.
- We can define interface with ***interface*** keyword.
- A class extends another class, an interface extends another interface but a **class implements an interface**.
- We can create Object reference for Interface but we cannot instantiate interface.



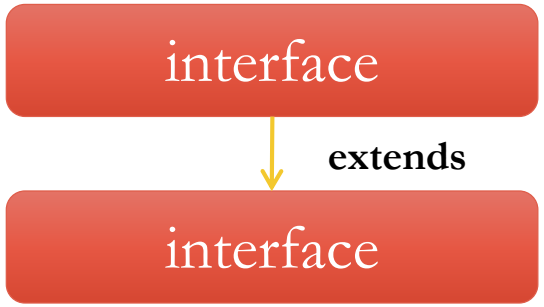
Java Interface




```
class A
{
//Variables
//Methods
}
class B extends A
{
//Variables
//Methods
}
```

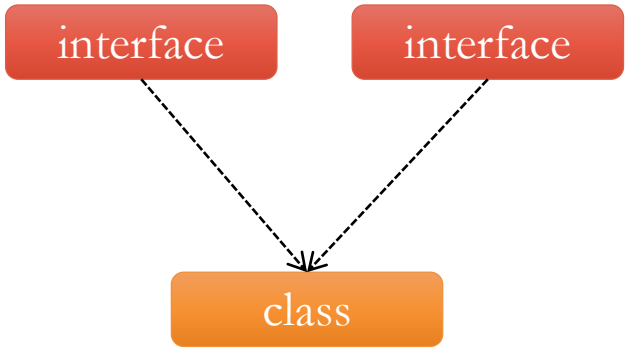


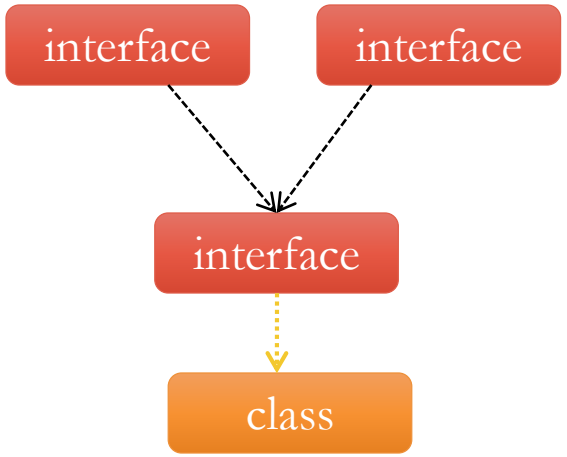

```
interface I
{
//abstract methods
//final static variables
}
class B implements I
{
//Method implementation;
}
```




```
interface I1
{
//abstract methods
//final static variables
}
Interface I2 extends I1
{
//abstract methods
//final static variables
}
class B implements I2
{
//Methods implementation;
}
```

Multiple Inheritance in Java by Interface

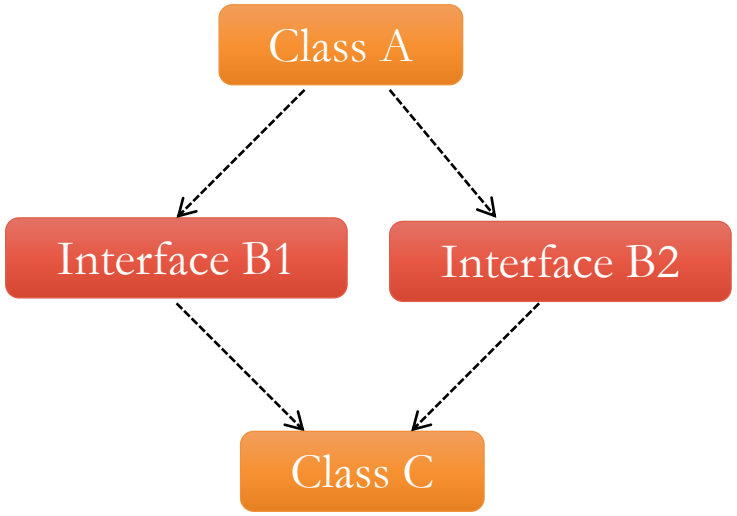





```
class A implements I1, I2
{
//Implement all the methods from I1 & I2
}
```

```
interface I extends I1, I2
{
//final static variables
//abstract methods
}
class A implements I
{
//Implement all the methods from I
}
```

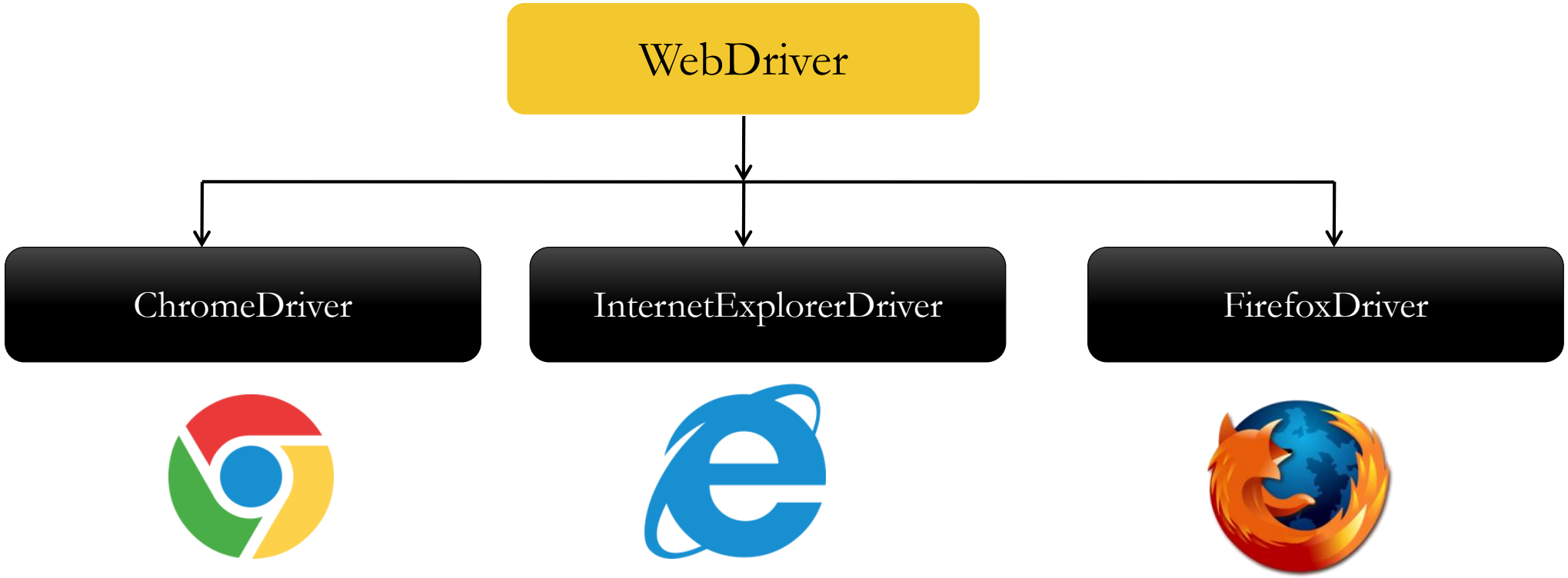
Hybrid inheritance in java by interface



class C **extends** A **implements** B1, B2
{
    //Implements methods from B1 & B2
}

```
class A1 {
void m1() {
System.out.println(" This is m1 from class A1");
}
}

interface B1 {
void m2();
}

interface B2 {
void m3();
}

class C extends A1 implements B1, B2 {
public void m2() {
System.out.println(" This is m2 from interface B1");
}

public void m3() {
System.out.println(" This is m3 from interface B2");
}
}

public class Test4 {
public static void main(String[] args) {

C cobj = new C();
cobj.m1();
cobj.m2();
cobj.m3();
}
}
```

Selenium WebDriver is an interface



Java Packages

- A **java package** is a group of similar types of classes, interfaces and sub-packages.
- Package in java can be categorized in two forms.
  - Built-in package
  - User-defined package
- There are many built-in packages such as java, lang, awt, javax, swing, net, io, util, sql etc.

Access package from another package

- There are two ways to access the package from outside the package.
  - import package.*;
  - import package.classname;

Access Modifiers in java

- The access modifiers in java specifies accessibility (scope) of a data member, method, constructor or class.
- There are 4 types of java access modifiers:
  - private
  - default
  - protected
  - public

private access modifier

```
class A{
private int data=40;
private void msg(){
System.out.println("Hello java");
}
}

public class Simple{
 public static void main(String args[]){
   A obj=new A();
   System.out.println(obj.data);//Compile Time Error
   obj.msg();//Compile Time Error
   }
}
```

default access modifier

- If you don't use any modifier, it is treated as **default** by default. The default modifier is accessible only within package.

```
//save by A.java
package pack;
class A{
  void msg()
    {
    System.out.println("Hello");
    }
}
```

```
//save by B.java
package mypack;
import pack.*;
class B{
  public static void main(String args[]){
   A obj = new A();//Compile Time Error
   obj.msg();//Compile Time Error
  }
}
```

- * In the above example, the scope of class A and its method msg() is default so it cannot be accessed from outside the package.

protected access modifier

- The **protected access modifier** is accessible within package and outside the package but through inheritance only.
- The protected access modifier can be applied on the data member, method and constructor. It can't be applied on the class.

```
//save by A.java
package pack;
public class A
{
protected void msg()
{
System.out.println("Hello");
}
}
```

```
//save by B.java
package mypack;
import pack.*;

class B extends A
{
  public static void main(String args[])
  {
   B obj = new B();
   obj.msg();
  }
}
```

public access modifier

- The public access modifier is accessible everywhere. It has the widest scope among all other modifiers.

```
//save by A.java

package pack;
public class A{
public void msg()
{System.out.println("Hello");
}
}
```

```
//save by B.java

package mypack;
import pack.*;

class B{
  public static void main(String args[])
 {
   A obj = new A();
   obj.msg();
  }
}
```

Access modifiers

Access Modifier	within class	within package	outside package by subclass only	outside package
Private	Y	N	N	N
Default	Y	Y	N	N
Protected	Y	Y	Y	N
Public	Y	Y	Y	Y

Assignment

1. Write a program to demonstrate interface.
   - Interface A : int a, int b sum()
   - Class B : Implements method from A and calculate sum of a and b

- 2. Write a program for multiple inheritance by using interface.
   - Interface A : int a, int b add()
   - Interface B : int x, int y mul()
   - Class Calculation : Implements methods from A and B interfaces.