# Interface

* An interface in Java is a blueprint of a class. It has static constants and abstract methods.
* The interface in Java is *a mechanism to achieve****abstraction***. There can be only abstract methods in the Java interface, not method body. It is used to achieve abstraction and multiple inheritance in java.
* By default, every variable in interface is final and static
* Note: Static variable is common to all the instances in the class
* Interface cannot be instantiated
* Interface don’t have own memory in the heap that’s why we can only have static and final variables there.
* Multiple inheritance is supported in interface
* Interface extend interface, Class implements interface

Interface A

{

void m1();

}

Class B implements A

{

Public void m1()

{

System.out.println(“This is defined”)

}

Public class ClassName

{

Public static void main(String[]args)

{

A obj;

Obj = new B();

}

}

# Need for interface

* It eliminates tight coupling
* Note: We can create reference of a parent class and object of a child class

## Sample code

package com.practice.fundamentals;  
  
interface Computer1{  
 void code();  
}  
class Lap implements Computer1{  
 public void code(){  
 System.*out*.println("Helps to code in moderate speed");  
 }  
}  
class Desk implements Computer1{  
 public void code(){  
 System.*out*.println("Helps to code faster");  
 }  
}  
class Person{  
 public void devApp(Computer1 comp){  
 comp.code();  
 }  
}  
public class Sample{  
 public static void main(String[] args)  
 {  
 Lap lap = new Lap();  
 Desk desk = new Desk();  
  
 Person obj = new Person();  
 obj.devApp(lap);  
 obj.devApp(desk);  
 }  
}