

Functional Interface: If any interface has only 1 abstract method. The number of static / default/ private methods does not matter.

@FunctionalInterface: Compile time check that it actually has only 1 abstract method.

Eg:

@FunctionalInterface

**interface** I1{

**void** m1();//1 abstract method

}

//To create an object, we generally first create a class which implements this interface and then we create an object of this class

class XYZ implements I1{

@Override

public void m1() {

System.out.println("M1 called");

}

}

XYZ obj= new XYZ();

//Anonymous inner class: Any interface or an abstract class

I1 obj= **new** I1() {//anonymous inner class

@Override

**public** **void** m1() {

System.***out***.println("m1 called");

}

};

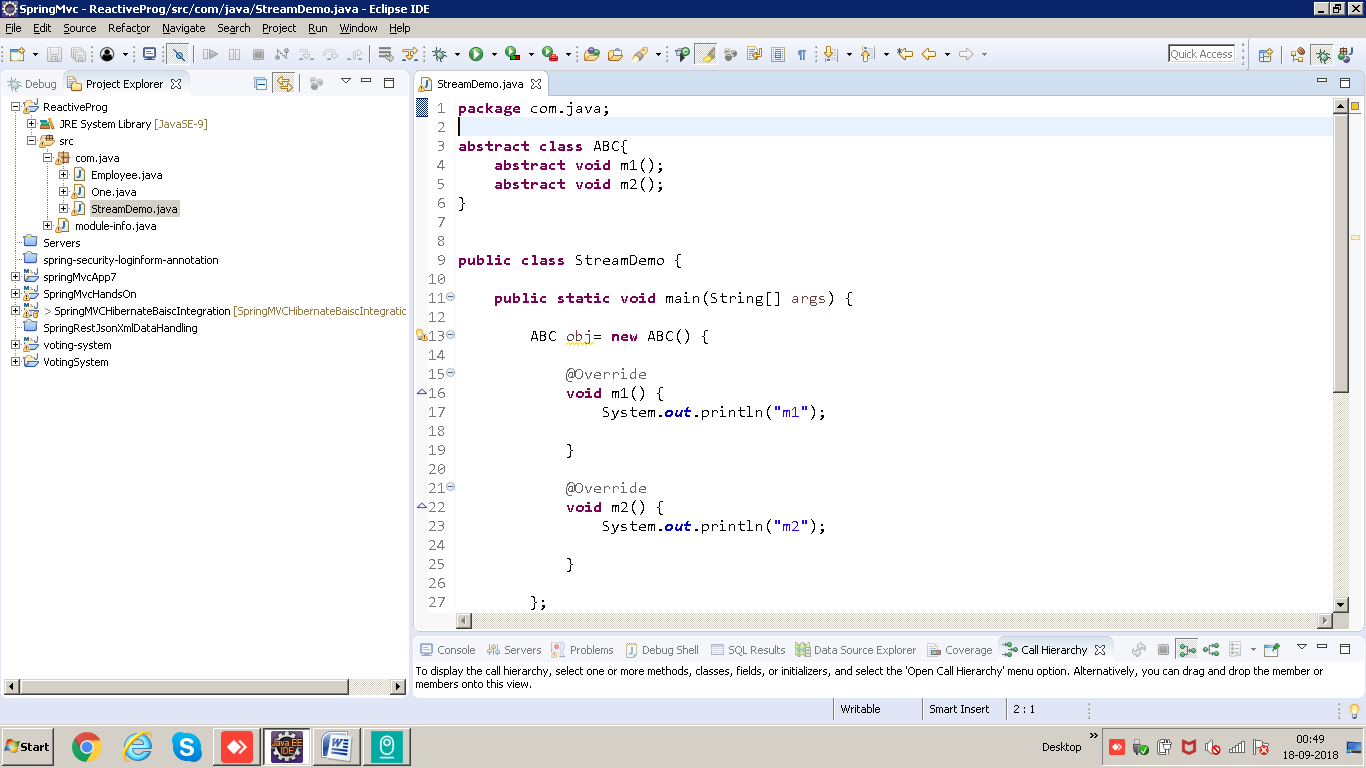
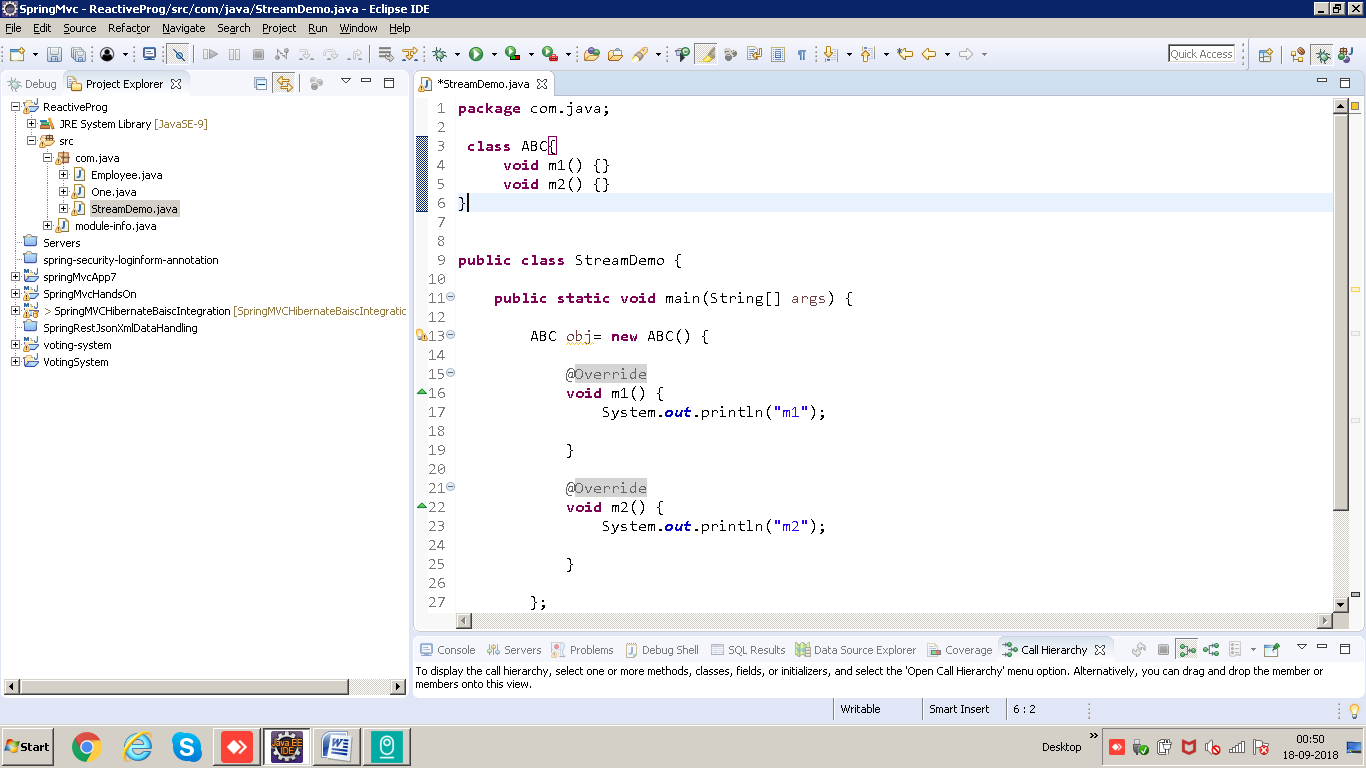
Eg2: Anonymous class for abstract class

**abstract** **class** ABC{

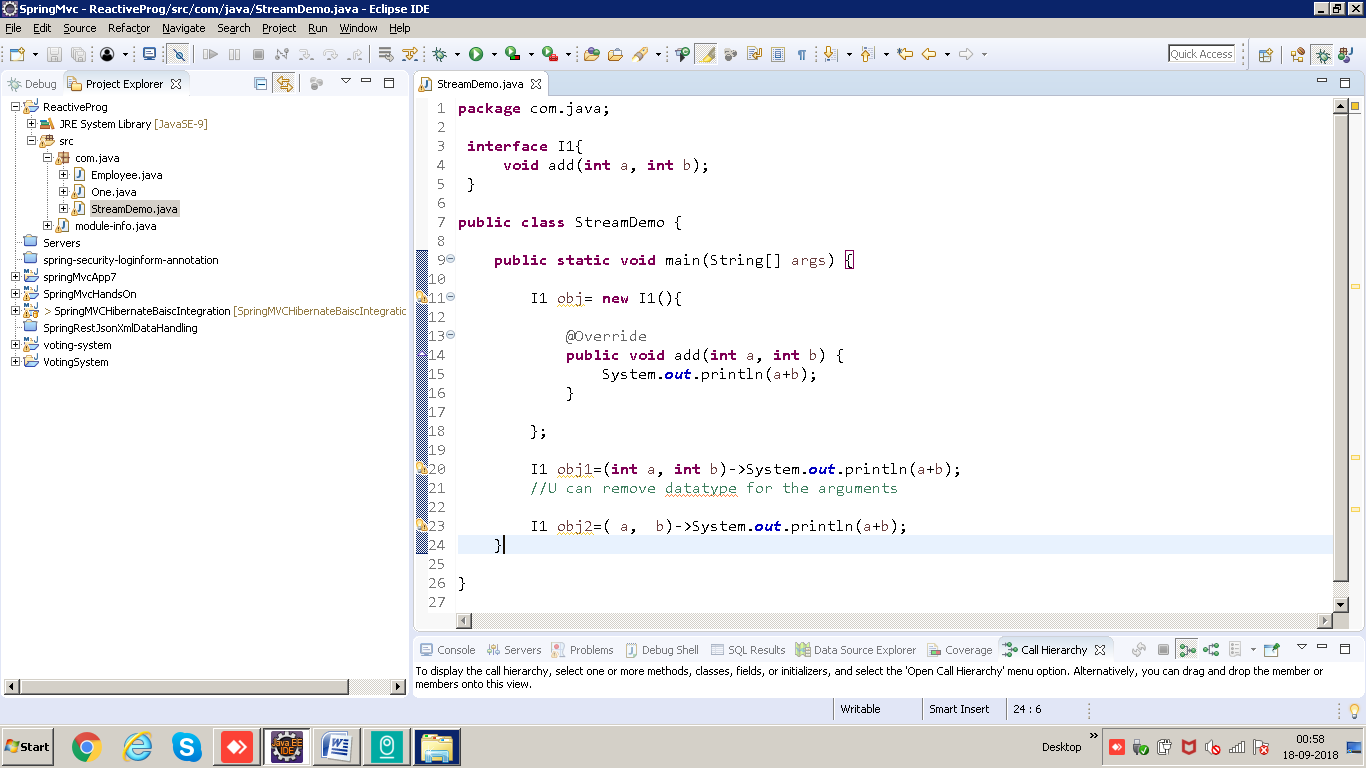
**abstract** **void** m1();

**abstract** **void** m2();

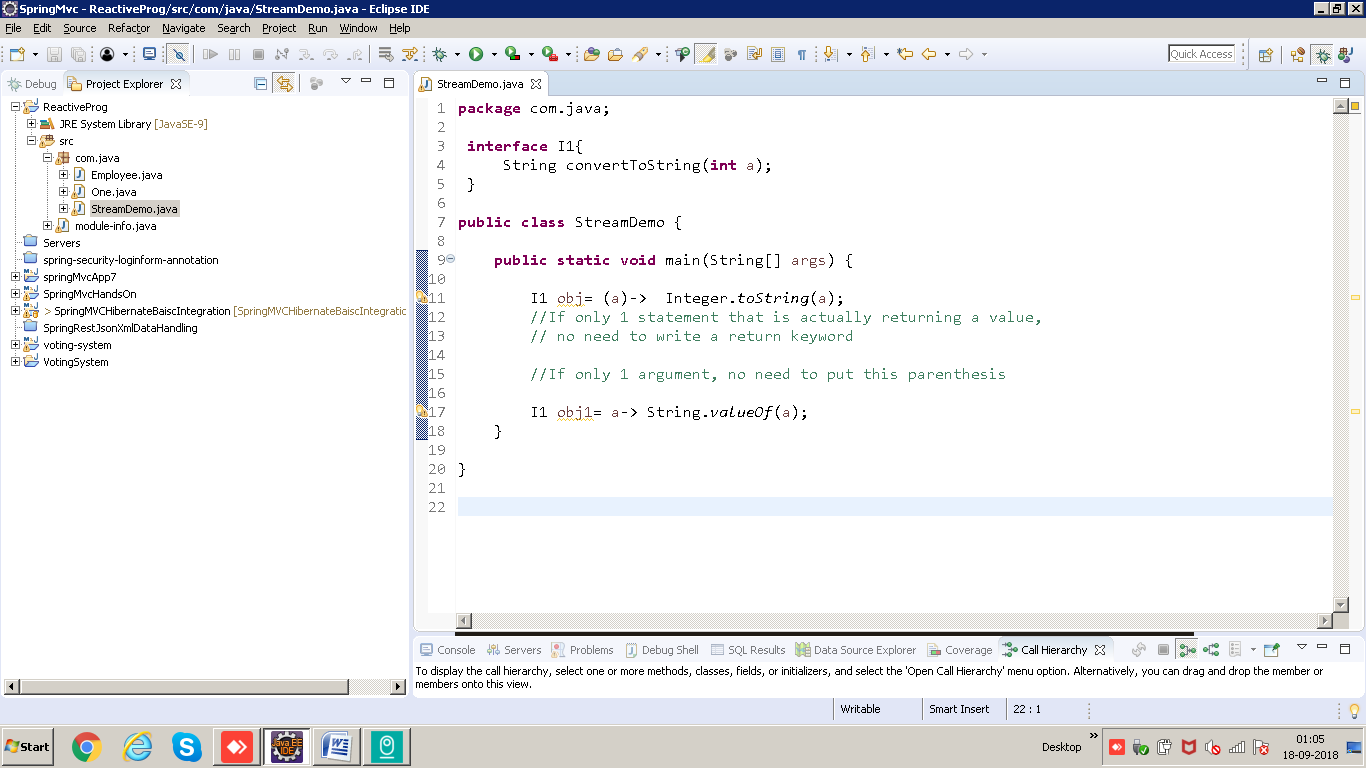
}

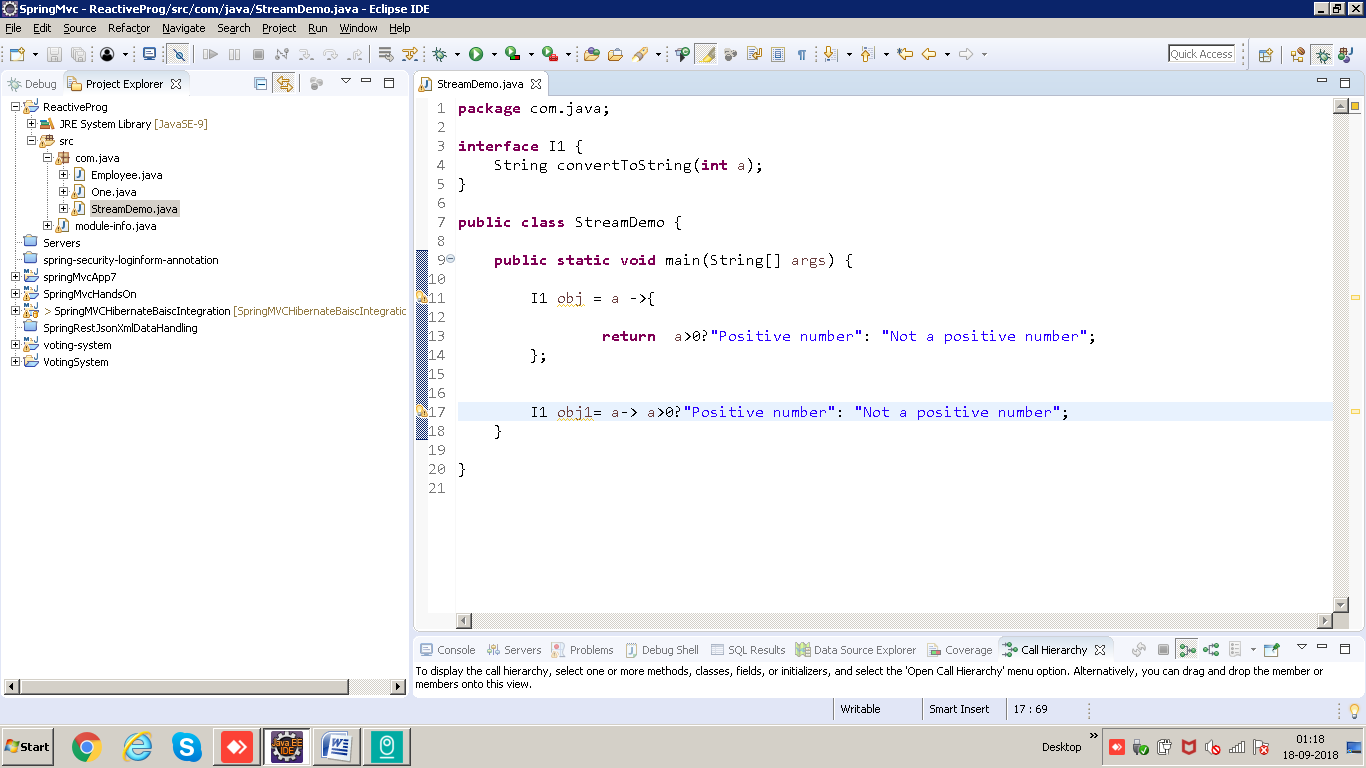
 

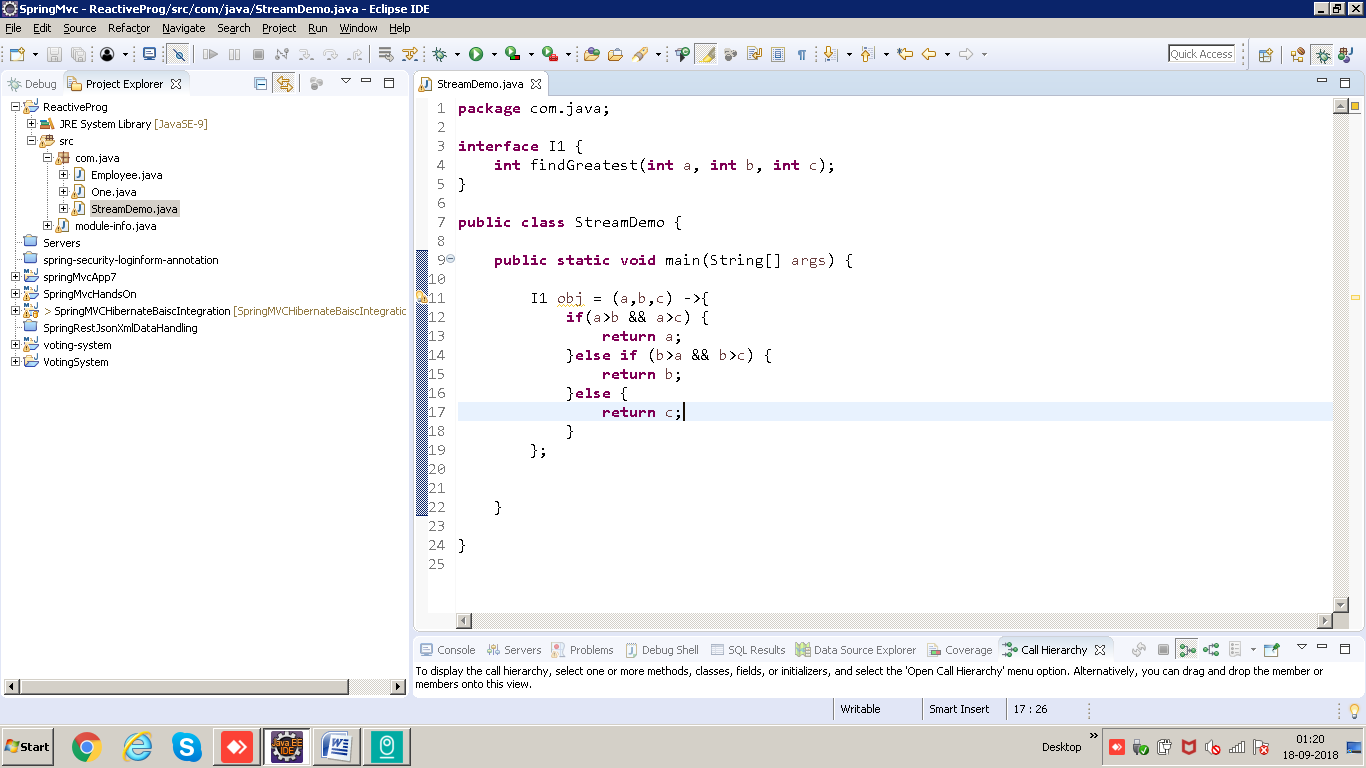
Lambda Expression: Works only for functional interface



@Override?



Example”



Using ternary operator:

