

## Scala Trait Mixins

In scala, trait mixins means you can extend any number of traits with a class or abstract class. You can extend only traits or combination of traits and class or traits and abstract class.

It is necessary to maintain order of mixins otherwise compiler throws an error.

You can use mixins in scala like this:

---

## Scala Trait Example: Mixins Order Not Maintained

In this example, we have extended a trait and an abstract class. Let's see what happen.

```
trait Print{  
  def print()  
}  
  
abstract class PrintA4{  
  def printA4()  
}  
  
class A6 extends Print with PrintA4{  
  def print(){           // Trait print  
    println("print sheet")  
  }  
}
```



```

def printA4(){           // Abstract class printA4
    println("Print A4 Sheet")
}

object MainObject{
    def main(args:Array[String]){
        var a = new A6()
        a.print()
        a.printA4()
    }
}

```

Output:

```

error: class PrintA4 needs to be a trait to be mixed in
class A6 extends Print with PrintA4{
                                ^
one error found

```

The above program throws a compile time error, because we didn't maintain mixins order.

## Scala Mixins Order

The right mixins order of trait is that any class or abstract class which you want to extend, first extend this. All the traits will be extended after this class or abstract class.

## Scala Trait Example: Mixins Order Maintained

```

trait Print{
    def print()
}

abstract class PrintA4{
    def printA4()
}

class A6 extends PrintA4 with Print{           // First one is abstract class second one is trait

```



```

def print(){                                // Trait print
    println("print sheet")
}

def printA4(){                              // Abstract class printA4
    println("Print A4 Sheet")
}

}

object MainObject{
    def main(args:Array[String]){
        var a = new A6()
        a.print()
        a.printA4()
    }
}

```

Output:

```

print sheet
Print A4 Sheet

```

## Another Example of Scala Trait

Here, we have used one more approach to extend trait in our program. In this approach, we extend trait during object creation. Let's see an example.

```

trait Print{
    def print()
}

abstract class PrintA4{
    def printA4()
}

class A6 extends PrintA4 {
    def print(){                                // Trait print
        println("print sheet")
    }

    def printA4(){                              // Abstract class printA4
        println("Print A4 Sheet")
    }
}

```



```

}

object MainObject{
  def main(args:Array[String]){
    var a = new A6() with Print    // You can also extend trait during object creation
    a.print()
    a.printA4()
  }
}

```

Output:

```

print sheet
Print A4 Sheet

```

← Prev

Next →

 For Videos Join Our Youtube Channel: [Join Now](#)

Help Others, Please Share




**IntelliPaat**

# Data Science Master Program

In collaboration with **IBM**













12 Courses   47 Projects   200 Hours

[ENROLL NOW](#)






Icons: TensorFlow, R, S, Python, and others.

Learn Latest Tutorials















SoapUI	RPA	Manual T.	Cucumber
 Appium	 PostgreSQL	 Solr	 MongoDB
 Gimp	 Verilog	 Teradata	 PhoneGap






















## Preparation

 Aptitude Aptitude	 Logical Reasoning Reasoning	 Verbal Ability Verbal A.	 Interview Questions Interview
 Company Interview Questions Company			

## Trending Technologies

 Artificial Intelligence Tutorial AI	 AWS Tutorial AWS	 Selenium tutorial Selenium	 Cloud tutorial Cloud
 Hadoop tutorial Hadoop	 ReactJS Tutorial ReactJS	 Data Science Tutorial D. Science	 Angular 7 Tutorial Angular 7
 Blockchain Tutorial Blockchain	 Git Tutorial Git	 Machine Learning Tutorial ML	 DevOps Tutorial DevOps

# B.Tech / MCA

 DBMS tutorial DBMS	 Data Structures tutorial DS	 DAA tutorial DAA	 Operating System tutorial OS
 Computer Network tutorial C. Network	 Compiler Design tutorial Compiler D.	 Computer Organization and Architecture COA	 Discrete Mathematics Tutorial D. Math.
 Ethical Hacking Tutorial E. Hacking	 Computer Graphics Tutorial C. Graphics	 Software Engineering Tutorial Software E.	 html tutorial Web Tech.
 Cyber Security tutorial Cyber Sec.	 Automata Tutorial Automata	 C Language tutorial C	 C++ tutorial C++
 Java tutorial Java	 .Net Framework tutorial .Net	 Python tutorial Python	 List of Programs Programs
 Control Systems tutorial Control S.	 Data Mining Tutorial Data Mining		