

Scala Singleton and Companion Objects

Last Updated : 15 Apr, 2021

Singleton Object

Scala is more object oriented language than Java so, Scala does **not** contain any concept of **static** keyword. Instead of static keyword Scala has **singleton object**. A Singleton object is an object which defines a single object of a class. A singleton object provides an entry point to your program execution. If you do not create a singleton object in your program, then your code compile successfully but does not give output. So you required a singleton object to get the output of your program. A singleton object is created by using **object** keyword.

Syntax:

```
object Name{  
  // code...  
}
```

Important points about singleton object

- The method in the singleton object is globally accessible.
- You are not allowed to create an instance of singleton object.
- You are not allowed to pass parameter in the primary constructor of singleton object.
- In Scala, a singleton object can extend class and traits.
- In Scala, a main method is always present in singleton object.
- The method in the singleton object is accessed with the name of the object (just

like calling static method in Java), so there is no need to create an object to access this method.

Example 1:

Scala

```
// A simple Scala program to illustrate
// the concept of singleton object

class AreaOfRectangle
{

    // Variables
    var length = 20;
    var height = 40;

    // Method which gives the area of the rectangle
    def area()
    {
        var ar = length * height;
        println("Height of the rectangle is:" + height);
        println("Length of the rectangle is:" + length);
        println("Area of the rectangle is :" + ar);
    }
}

// singleton object
object Main
{
    def main(args: Array[String])
    {

        // Creating object of AreaOfRectangle class
        var obj = new AreaOfRectangle();
        obj.area();
    }
}
```

Output:



Height of the rectangle is:40

Length of the rectangle is:20

Area of the rectangle is :800

Example 2:

Scala

```
// A Scala program to illustrate
// how to call method inside singleton object

// Singleton object with named
// as Exampleofsingleton
object Exampleofsingleton
{

    // Variables of singleton object
    var str1 = "Welcome ! GeeksforGeeks";
    var str2 = "This is Scala language tutorial";

    // Method of singleton object
    def display()
    {
        println(str1);
        println(str2);
    }
}
```



```
// Singleton object with named as Main
object Main
{
    def main(args: Array[String])
```



Related Articles

Save for later

```
}
```

Output:

```
Welcome ! GeeksforGeeks
This is Scala language tutorial
```

Explanation: In the above example, we have two singleton objects, i.e, Exampleofsingleton and Main. Exampleofsingleton object contains a method named as display(), now we call this method in Main object. Using this statement Exampleofsingleton.display(); we call display() method that is present in Exampleofsingleton object and print output.

Companion Object

Companion object is known as an object whose name is same as the name of the class. Or In other words, when an object and a class have the **same name**, then that object is known as the companion object and the class is known as companion class. A companion object is defined in the same source file in which the class is defined. A companion object is allowed to access both private methods and private fields of the class.

Example:

Scala

```
// A Scala program to illustrate
```



```
// the concept of the Companion object

// Companion class
class ExampleofCompanion
{

    // Variables of Companion class
    var str1 = "GeeksforGeeks";
    var str2 = "Tutorial of Companion object";

    // Method of Companion class
    def show()
    {
        println(str1);
        println(str2);
    }
}

// Companion object
object ExampleofCompanion
{
    def main(args: Array[String])
    {
        var obj = new ExampleofCompanion();
        obj.show();
    }
}
```

Output:

```
GeeksforGeeks
Tutorial of Companion object
```

Like 0

Previous

Operators in Scala

Next

Scala Constructors

RECOMMENDED ARTICLES

Page : 1 2 3

01 **Scala | Package Objects**
29, Apr 19

05 **Scala | Case Class and Case Object**
27, Feb 19

02 **Scala Tutorial – Learn Scala with Step By Step Guide**
25, Nov 19

06 **Scala | Format and Formatted Method**
29, Mar 19

03 **Scala SortedMap addString() method with a start, a separator and an end**

7 **Scala | Null, null, Nil, Nothing, None, and Unit**

with example

15, Oct 19

10, Apr 19

04 Difference Between Traits and Abstract Classes in Scala

13, Feb 19

08 Scala Console | println, printf and readLine

10, Apr 19

Article Contributed By :



[ankita_saini](#)

@ankita_saini

Vote for difficulty

Easy

Normal

Medium

Hard

Expert

Improved By : [arorakashish0911](#)

Article Tags : [Scala](#), [Scala-OOPS](#), [Scala](#)

Improve Article

Report Issue

Writing code in comment? Please use ide.geeksforgeeks.org, generate link and share the link here.

Load Comments



Geeks for Geeks

Company

About Us

Careers

Privacy Policy

Contact Us

Copyright Policy

Practice

Courses

Company-wise

Topic-wise

How to begin?

Learn

Algorithms

Data Structures

Languages

CS Subjects

Video Tutorials

Contribute

Write an Article

Write Interview Experience

Internships

Videos