
SCALA OOPS(CLASS AND OBJECTS): HANDS ON LAB

By www.HadoopExam.com

Contact us: hadoopexam@gmail.com / admin@hadoopexam.com

Note: These instructions should be used with the HadoopExam Scala: Professional Trainings. Where it is executed and you can do hands on with trainer.

One year subscription for following training at 50% Discount (Access any of the below training)

~~Actual Cost \$499/35999INR~~

After Discount \$249/17999INR

1. [Hadoop Professional Training](#)
2. [Spark Professional Training](#)
3. [HBase\(NoSQL\) Professional Training](#)
4. [Apache Oozie \(Hadoop workflow \) Training](#)
5. [Scala Professional Training](#)
6. [Python Programming Training](#)
7. [Java 1Z0-808 Core Java Programming Training](#)

Topics

1. Scala classes and objects
2. Constructor fields visibility
3. Auxiliary constructor
4. Lazy Variables
5. Option variable
6. Inheritance and abstract class
7. Case classes

Defining an Employee class with name and its id

```
class Employee (val id:Int, val name:String){ //Defining constructor
val e1 = new Employee(100, "Amit") //Creating Instance
val e2 = new Employee(101, "Rakesh") //Creating Instance
//Accessing values
e1.name
e2.name
```

Doing some activity, when new object is created.

```
class Employee (val id:Int, var name:String){
    private val company="HE.com"
    val fixedBonus=1000
    println(s"\n Employee Name : ${name} ID : ${id} Company : ${company} Fixed Bonus :
    ${fixedBonus} \n" )
}
val e1 = new Employee(100, "Amit")
val e2 = new Employee(101, "Rakesh")
e2.name = "Rajnish"
e2.id=104 //Error
e2.company //Not Accessible
e2.fixedBonus //Accessible
```

Example with val, var, private and no modifiers.

```
class Employee(var id:Int, val name:String, bonus:Int, private var fullSalary:Int){
    | def getFullSalary {
    | | println(fullSalary)
    | }
    | }

val e = new Employee(100,"Amit",1000,80000)

e.id
e.name
e.bonus
e.fullSalary

e.id=101
e.name="Amit Kumar"
e.bonus=2000
e.fullSalary=85000

e.getFullSalary
```

Auxiliary Constructor: Multiple ways to create instance of a Class (With or without pre-assigned values)

```
class Employee(var id:Int, var name:String, var bonus:Int, var fullSalary:Int){
  def this(id:Int,name:String){
    this(id,name,1000,50000)
  }

  def this(){
    this(0,"",1000,50000)
  }

  def info = s"Employee Detail ID: $id , Name : $name , Bonus : $bonus , Salary :$fullSalary"
}

val e1=new Employee()
e1.info

val e2=new Employee(101, "Rakesh")
e2.info

val e3=new Employee(102, "Amit Kumar" , 5000 , 95000)
e3.info
```

Creating Class with default values

```
class Employee(var id:Int=0, var name:String="", var bonus:Int=1000, var fullSalary:Int=5000){

  def info = s"Employee Detail ID: $id , Name : $name , Bonus : $bonus , Salary :$fullSalary"
}

val e1 = new Employee()
e1.info

val e2 = new Employee(101, "Amit Kumar")
e2.info

val e3 = new Employee(101, "Amit Kumar" , 5000, 85000)
e3.info
```

Lazy variables

```
class Employee( var name:String="", var bonus:Int=1000, var fullSalary:Int=5000){
  lazy val id={
```

```
        if(name=="Amit")
            101
        else
            0
    }
    def info = s"Employee Detail ID: $id , Name : $name , Bonus : $bonus , Salary :$fullSalary"
}

val e1 = new Employee()
e1.info

val e1 = new Employee("Amit")
e1.info
```

Assigning Default Values (Other than Core types)

```
case class PrevEmployer (val prevSalar:Int, val name:String)

class Employee( var id:Int=0,var name:String="", var bonus:Int=1000, var fullSalary:Int=5000){
    var prevEmployer = None:Option[PrevEmployer]
    def info = s"Employee Detail ID: $id , Name : $name , Bonus : $bonus , Salary :$fullSalary "
}

val e1 = new Employee()
e1.info
e1.prevEmployer

e1.prevEmployer=Some(PrevEmployer(40000, "QuickTechie"))
e1.prevEmployer.foreach(prev => print(prev.name))
```

Extending Class

- Parent class is a Person and child class is an Employee.
- Person class has two variables: name and age
- Employee class will have two additional parameters: bonus and salary
- Declare your base class as usual with val or var constructor parameters
- When defining a subclass constructor, leave the val or var declaration off of the fields that are common to both classes.
- Then define new constructor parameters in the subclass as val or var fields, as usual.

```
class Person(var name:String, var age:Int)
class Employee (name:String, age:Int, var bonus:Int, var fullSalary:Int) extends Person (name, age){
    def info = s"Employee Detail Name : $name , Bonus : $bonus , Salary :$fullSalary "
}
val e1 = new Employee("Amit",30,5000,85000)
e1.info
```

```
class Person(var name:String, var age:Int) {
    def this(name:String){
        this(name,21)
    }
}

class Employee (name:String, age:Int, var bonus:Int, var fullSalary:Int) extends Person (name, age){

    def this (name:String){
        this(name,21,1000,50000)
    }

    def info = s"Employee Detail Name : $name , Bonus : $bonus , Salary :$fullSalary "
}

//Another way to extend Person class
class Employee (name:String, age:Int, var bonus:Int, var fullSalary:Int) extends Person (name){

    def this (name:String){
        this(name,21,1000,50000)
    }

    def info = s"Employee Detail Name : $name , Bonus : $bonus , Salary :$fullSalary "
}

val e1 = new Employee("Amit")
e1.info
```

Abstract Class:

```
abstract class Person(var name:String, var age:Int) {
    def calculateTotalSalary
}

class Employee (name:String, age:Int, var bonus:Int, var fullSalary:Int) extends Person (name, age){
    def info = s"Employee Detail Name : $name , Bonus : $bonus , Salary :$fullSalary "
}

class Employee (name:String, age:Int, var bonus:Int, var fullSalary:Int) extends Person (name, age){
    def calculateTotalSalary = println(bonus+fullSalary)
    def info = s"Employee Detail Name : $name , Bonus : $bonus , Salary :$fullSalary "
}

val e1 = new Employee("Amit",30,5000,85000)
e1.calculateTotalSalary
```

Case Classes

```
case class Employee (name:String, age:Int, var bonus:Int, var fullSalary:Int)
val e1=Employee(name="Rakesh",age=30,bonus=5000,fullSalary=85000)
val e2=e1.copy(name="Amit",age=30,bonus=5000,fullSalary=85000)
```

One year subscription for following training at 50% Discount (Access any of the below training)

Actual Cost ~~\$499/35999INR~~

After Discount \$249/17999INR

- | | |
|--|---|
| 1. Hadoop Professional Training | 2. Spark Professional Training |
| 3. HBase(NoSQL) Professional Training | 4. Apache Oozie (Hadoop workflow) Training |
| 5. Scala Professional Training | 6. Python Programming Training |
| 7. Java 1Z0-808 Core Java Programming Training | |

Customize Your Package : Following Products are available. You can customize your own package. Please select Products from below list and send an email to hadoopexam@gmail.com . So, our team will reply with discounted price.

Recorded Trainings (With Hands On Lab) 1. Hadoop BigData Professional Training (3500INR/\$79) 2. HBase (NoSQL) Professional Training (3500INR/\$79) 3. Apache Spark Professional Training (3900INR/\$89 for a week 3500INR/\$75) 4. Apache Oozie (Hadoop Workflow Engine) Training	5. Apache Spark Oreilly Developer Certification Hortonworks Certification : 6. HDPCD : No Java	EMC :Data Science Certification Practice Questions 7. EMC E20-007 8. EMC E20-065
Cloudera Certification Practice Question Bank 9. CCA50X : Hadoop Administrator 10. CCA-175 Cloudera® (Hadoop and Spark Developer) 11. CCP:DE575 : Cloudera® Data Engineer Certification 12. CCA159 : Cloudera® Data Analyst Certification	AWS : Amazon WebService Certification Practice Question 13. AWS Solution Architect : Associate 14. AWS Solution Architect: Professional 15. AWS Developer : Associate 16. AWS Sysops Admin : Associate	SAS Certffication Practice Questions 17. SAS Base A00-211 18. SAS Advanced A00-212 19. SAS Analytics : A00-240 20. SAS Administrator : A00-250
MapR Hadoop/BigData Certification Practice Questions 21. MapR Hadoop Developer Certification 22. MapR HBase NoSQL Certification 23. MapR Spark Developer Certification	Microsoft Azure Cloud Certification 24. Azure 70-532 25. Azure 70-533	Oracle Certification Practice Questions 26. Java 1z0-808 27. 1z0-060 (Oracle 12c) 28. 1z0-061 (Oracle 12c)

DON'T FORGET TO SUBSCRIBE FOR UPDATE ON PRODUCTS

[Subscribe](#)

[Request an Exam or Training](#)

HadoopExam Learning Resource provides the following material for the Advanced Technologies.
Please visit www.HadoopExam.com for more detail this is just a few products from portfolio.

Price start for training with Just \$79/3500INR



Apache Spark
Professional
Training with
HandsOn Session

+ Certification
Material



Hadoop Professional
Training with
HandsOn Session

+ Certification
Material



HBase Professional
Training with
HandsOn Session

+ Certification
Material



Certification
Material



Certification
Material



Certification
Material



Certification
Material



Certification
Material



Microsoft Azure

Certification
Material



Certification
Material