1. **Comparative study**

|  |  |
| --- | --- |
| **Java** | **Scala** |
| **A Book Class in Java**  class Book{  private String title;  private int numberOfPages;  public Book(String title, int numberOfPages) { this.title = title;  this.numberOfPages = numberOfPages;  }  } | **A Book Class in Scala**  class Book(title: String, numberOfPages: Int) |
| **Java Equivalent of Book POJO**  public class Book implements scala.Product,scala.Serializable {  public static scala.Option<scala.Tuple2>unapply(Book); public static Book apply(java.lang.String, int); public static scala.Function1<scala.Tuple2, Book> tupled(); public static scala.Function1>curried(); public java.lang.String title(); public void title\_$eq(java.lang.String); public int numberOfPages(); public void numberOfPages\_$eq(int); public Book copy(java.lang.String, int); public java.lang.String copy$default$1(); public int copy$default$2(); public java.lang.String productPrefix(); public int productArity(); public java.lang.Object productElement(int); public scala.collection.Iterator productIterator(); public boolean canEqual(java.lang.Object); public int hashCode(); public java.lang.String toString(); public boolean equals(java.lang.Object); public Book(java.lang.String, int); } | **Lightweight Book POJOin Scala**  case class Book(var title: String, var numberOfPages: Int) |
| **Static member in Java** | **Singleton Object in Scala object**  HelloWorld {  def greet() {  println("Hello World!")  }  } |
|  |  |

1. **Functional programming**

A Simple square Function

(i: Int) => { i \* i }

**Assign a function to a variable ->**

val square = (i: Int) => { i \* i }

1. **Scala REPL**

|  |  |
| --- | --- |
| **Command** | **output** |
| println("Hello World!"); | Hello World! |
| 1 + 1 | res1: Int = 2 |
| res1 \* 8 | res2: Int = 16 |
| val x = "Hello World" | x: String = Hello World |
| var xl = x.length | x1: Int = 11 |
| import java.util.\_ | import java.util.\_ |
| val d = new Date | d: java.util.Date = Fri Sep 11 14:52:53 IST 2015 |
| :paste  val v = 5  if (v==5)  print("true ")  else  print("false ") | // Entering paste mode (ctrl-D to finish)  …………..  ………….  …………  ………….  // Exiting paste mode, now interpreting.  true v: Int = 5 |
|  |  |

**NOTE:-** having java.util library in session which we can use.

1. **Writing scala programs-**

***“Hello world” program-***

object HelloWorld {

def main(args: Array[String]) {

println("Hello,World!")

}

}

*Execution- scala helloWorld.scala*

**Printing some numbers-**

**print1.scala**

for {i <- 1 to10}

println(i)

Note: lt prints the numbers from 1 to 10.

**print2.scala**

for {i <- 1 to10

j <- 1 to10}

println(i\* j)

*Note: In this program, we are iterating over 1 to 10 in an outer loop and assigning each number to i. In the inner loop, we are also iterating from 1 to 10 and assigning each number to j. The product of i \* j is printed, so you’ll see 10 lines output.*

***Note: Written & Executed programs at*** *C:\scala\_book\*