

```
one.scala × two.scala three.scala four.scala
one.scala
1  object one{
2      def a (k: Int, m: Int): Int = k+(12*m)
3
4      def b (m: Int, j: Int): Double = m/j
5
6      def c1 (n: Int, j: Int): Double = n%j
7
8      def d1 (m: Int, j: Int): Double = m/j*j
9
10     def e (f: Double, g: Double): Double = f+10*5+g
11
12     def f1(i: Int, n: Int): Int =(i+1)*n
13
14     def main(args: Array[String]): Unit = {
15         var (k,i,j) = (2,2,2)
16         var (m,n)=(5,5)
17         var f :Double=12.0
18         var g :Double=4.0
19         var c='X'
20
21         println(a(k,m))
22         println(b(m,j))
23         println(c1(n,j))
24         println(d1(m,j))
25         println(e(f,g))
26         println(f1(i,n))
27     }
28 }
```

62
2.0
1.0
4.0
66.0
15

- Scala's syntax is more expressive than Java and lets programmers use compact code.
- While both languages supports OOP scala is able to combine its functional programming concepts with OOP. Java also includes functional programming support but its mainly treated as an OOP language.
- Both has a static stype system but sometimes scala is flexible so the compiler can deduce types.
- Java has built-in multithreading and concurrent programming support. While scala provides higher-level abstractions for concurrency and parallelism. Also scalas's immuatble data structures and types provide 'safe concurrent programming'
- Java is widely used in enterprise applications while scala focuses more on scaleable and functional programming.

```

one.scala  two.scala ×  three.scala  four.scala
two.scala
1  object two{
2      def inc(x: Int): Int = x+1
3      def dec(x: Int): Int = x-1
4
5      def a1(b: Int, a: Int, c: Int, d: Int): Int = (b*a) + (c*d)
6
7      def c1(g: Float, k: Float, c: Int): Float = -2*(g-k)+c
8
9      def main(args: Array[String]): Unit = {
10
11
12          var (a,b,c,d) = (2,3,4,5)
13          var k: Float = 4.3f
14          var g :Float=4.0f
15
16          //b=inc(b)
17          println(a1(dec(b),a,c,dec(d)))
18          println(inc(a))
19          println(c1(g,k,c))
20          c=inc(c); println(c)
21          c=inc(c)*inc(a); println(c)
22
23
24      }
25
26 }

```

20
3
4.6000004
5
18

```
one.scala two.scala three.scala × four.scala
three.scala
1 object three{
2   def salary(nhrs: Int, othrs: Int): Double = ((250*nhrs)+(othrs*85))*(1-0.12)
3   |
4   def main(args: Array[String]): Unit = {
5     |   println(salary(40,30))
6     |
7   }
```

11044.0

```
one.scala two.scala three.scala four.scala ×
four.scala
1 object four{
2   def revenue(price: Int, attendees: Int): Int = price*attendees
3
4   def cost(attendees: Int): Int = 500 + attendees*3
5
6   def profit(rev: Int, c: Int): Int = rev - c
7
8   def max(p1: Int, p2: Int, p3: Int): Int = {
9     if (p1 < p2){
10      |   if (p2 < p3) 10
11      |   else 20
12      |
13      else{
14      |   if (p1 > p3) 15
15      |   else 10
16      |
17    }
18
19    def main(args: Array[String]): Unit = {
20
21      val p1 = profit(revenue(15, 120), cost(120))
22      val p2 = profit(revenue(20, 100), cost(100))
23      val p3 = profit(revenue(10, 140), cost(140))
24
25      println("Best ticket price: " + max(p1, p2, p3))
26    }
27  }
```

Best ticket price: 20