

Functional Programming

Tutorial 2

01. Consider the following variables i, j, m, n, k, f, g, c. Declare the variables in Scala and assign with the initial values as follows: k = i = j = 2; m = n = 5; f = 12.0f; g = 4.0f; c = 'X';

Evaluate the following expressions:

a) $k + 12 * m$

b) m / j

c) $n \% j$

d) $m / j * j$

e) $f + 10 * 5 + g$

f) $++i * n$

```
scala> var k,i:Int=2
var k: Int = 2
var i: Int = 2

scala> var k,i,j:Int=2
var k: Int = 2
var i: Int = 2
var j: Int = 2

scala> var m,n:Int=5
var m: Int = 5
var n: Int = 5

scala> var f:Float=12.0f
var f: Float = 12.0

scala> var g:Float=4.0f
var g: Float = 4.0

scala> var c:Char='X'
var c: Char = X
```

```
scala> k+12*m
val res0: Int = 62

scala> m/j
val res1: Int = 2

scala> n%j
val res2: Int = 1

scala> m/j*j
val res3: Int = 4

scala> f+10*5+g
val res4: Float = 66.0

scala> i=i+1
i: Int = 3

scala> i*n
val res5: Int = 15
```

02. Use the following declaration and initialization to convert them to acceptable Scala statements. `int a = 2, b = 3, c = 4, d = 5;`

`float k = 4.3f;`

and evaluate the following expressions

a) `println(- -b * a + c *d - -);`

b) `println(a++);`

c) `println (-2 * (g - k) +c);`

d) `println (c=c++);`

e) `println (c=++c*a++);`

```
Microsoft Windows [Version 10.0.19045.3086]
(c) Microsoft Corporation. All rights reserved.

C:\Users\pathumi>scala
Welcome to Scala 3.3.0 (20.0.1, Java Java HotSpot(TM) 64-Bit Server VM).
Type in expressions for evaluation. Or try :help.

scala> var a:Int=2;
var a: Int = 2

scala> var b:Int=3;
var b: Int = 3

scala> var c:Int=4;
var c: Int = 4

scala> var d:Int=5;
var d: Int = 5

scala> var k:Float=4.3f;
var k: Float = 4.3

scala> var g:Float=4.0f;
var g: Float = 4.0
```

a)

```
scala> b=b-1  
b: Int = 2  
  
scala> println(b*a+c*d)  
24  
  
scala> d=d-1  
d: Int = 4
```

b)

```
scala> println(a)  
2  
  
scala> a=a+1  
a: Int = 3
```

c)

```
scala> println(-2*(g-k)+c)  
4.6000004
```

d)

```
scala> println(c)  
4  
  
scala> c=c+1  
c: Int = 5
```

e)

```
scala> c=c+1
c: Int = 6

scala> println(c*a)
18

scala> a=a+1
a: Int = 5
```

03. Company XYZ & Co. pays all its employees Rs.250 per normal working hour and Rs. 85 per OT hour. A typical employee works 40 (normal) and 30(OT) hours per week has to pay 12% tax. Develop a functional program that determines the take home salary of an employee from the number of working hours and OT hours given.

```
Type in expressions for evaluation. Or try :help.

scala> var normalHour:Int=250
var normalHour: Int = 250

scala> var OTHour:Int=85
var OTHour: Int = 85

scala> var tax:Float=0.12
var tax: Float = 0.12

scala> var noOfNormalHour:Int=40
var noOfNormalHour: Int = 40

scala> var noOfOTHour:Int=30
var noOfOTHour: Int = 30

scala> def feeForNormalHour(noOfNormalHour:Int):Int=noOfNormalHour*normalHour
def feeForNormalHour(noOfNormalHour: Int): Int

scala> def feeForOTHour(noOfOTHour:Int):Int=noOfOTHour*OTHour
def feeForOTHour(noOfOTHour: Int): Int

scala> def income():Int=feeForNormalHour(noOfNormalHour)+feeForOTHour(noOfOTHour)
def income(): Int

scala> def taxFee(tax:Float):Float=income()*tax
def taxFee(tax: Float): Float
```

```
scala> def takeHome():Float=income()-taxFee(tax)
def takeHome(): Float
```

```
scala> print("Take home sallary is Rs: " +takeHome())
Take home sallary is Rs: 11044.0~

scala> _
```

04. Imagine the owner of a movie theater who has complete freedom in setting ticket prices. The more he charges, the fewer the people who can afford tickets. In a recent experiment the owner determined a precise relationship between the price of a ticket and average attendance. At a price of Rs 15.00 per ticket, 120 people attend a performance. Decreasing the price by 5 Rupees increases attendance by 20 and increasing the price by 5 Rupees decreases attendance by 20. Unfortunately, the increased attendance also comes at an increased cost. Every performance costs the owner Rs.500. Each attendee costs another 3 Rupees. The owner would like to know the exact relationship between profit and ticket price so that he can determine the price at which he can make the highest profit. Implement a functional program to find out the best ticket price.

```
scala> def attendees(ticketprice:Int):Int=120+(15-ticketprice)/5*20
def attendees(ticketprice: Int): Int

scala> def cost(ticketprice:Int):Int=500+(3*attendees(ticketprice))
def cost(ticketprice: Int): Int

scala> def revenue(ticketprice:Int):Int=attendees(ticketprice)*ticketprice
def revenue(ticketprice: Int): Int

scala> def profit(ticketprice:Int):Int=revenue(ticketprice)-cost(ticketprice)
def profit(ticketprice: Int): Int
```

```
scala> def findBestTicketPrice(): Double = {
  |   var bestPrice: Double = 0
  |   var maxProfit: Double = Double.MinValue
  |
  |   for (ticketprice <- Range(15, 36, 5)) {
  |     val prof = profit(ticketprice)
  |     if (prof > maxProfit) {
  |       maxProfit = prof
  |       bestPrice = ticketprice
  |     }
  |   }
  |
  |   bestPrice
  | }
def findBestTicketPrice(): Double

scala> val bestPrice=findBestTicketPrice()
val bestPrice: Double = 25.0
```