SCALA TUTORIAL 2

20002025 – M.A.D.V.Weerasinghe

1. Consider the following variables required.

   i,j,m,n,k;
   f,g;
   c;
   then 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’;
   and 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,j:Int=2;
var m,n:Int=5;
var f=12.0f;
var g=4.0f;
var c='X';

@main
def run():Unit={
    q1()
    q2()
    q3()
    q4()
    q5()
    // q6()
}
// def ++(n:Int):Int=n+1
def q1():Unit=println(k+12*m);
def q2():Unit=println(m/j);
def q3():Unit=println(n%j);
def q4():Unit=println(m/j*j);
def q5():Unit=println(f/10*5+g);
// def q6():Unit=println(++(i)*n);
//++ Operator is not work in Scala and
// ++ is valid method
```

SCALA TUTORIAL 2

20002025 – M.A.D.V.Weerasinghe

2. Compare the Java and Scala programming languages.

Java	Scala
Not has a Lazy evaluation feature	Has lazy evaluation feature
Concise code type	Compact code type
Readability is high	Readability is low
Not Support operator overloading	Support Operator overloading
Backward Compatible	Not Backward Compatible

3. 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++);

```scala
var (a,b,c,d)=(2,3,4,5)
var k:Float=4.3f;

@main
def run():Unit={
    q1()
    q2()
    a=Inc(a)
    // q3()
    q4()
    c=Inc(c)
    q5()
}
def Inc(n:Int):Int=n+1
def --(n:Int):Int=n-1
def q1():Unit=println(--(b)*a +c*d);
//d-- is not a valid operator in scala.
//But since this is postfix decrement it won't affect to answer
def q2():Unit=println(a);
//a++ is not an valid operator in scala.
//But But since this is postfix decrement it won't affect to answer
// def q3():Unit=println(-2+(g-k)+c);
def q4():Unit=println(Inc(c)); //c++ is not a valid operator
def q5():Unit=println(Inc(c)*a);
//++,-- Operator is not work in Scala and
// ++,-- is valid method
```

SCALA TUTORIAL 2

20002025 – M.A.D.V.Weerasinghe

4. Write Scala functions to solve the following problems.
   a. 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 must 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.

```scala
import scala.io.StdIn.readInt
@main
def Run():Unit={
    println("Enter the Normal Hours")
    val Normal_Hours=readInt();
    println("Enter the OT Hours")
    val OT_Hours=readInt();
    val pureSalary= PureSalary(Normal_Hours,OT_Hours)
    val tax=Tax(Normal_Hours,OT_Hours)
    println(Salary(pureSalary,tax));
}
def PureSalary(Normal_Hours:Int,OTHours:Int):Int = {250*Normal_Hours + 85*OTHours}
def Tax(NormalHrs:Int,OTHrs:Int):Float=if(NormalHrs>=40 && OTHrs>=30) 0.12 else 0
def Salary(PureSalary:Int,Tax:Float):Float={PureSalary-PureSalary*Tax}
```

SCALA TUTORIAL 2

20002025 – M.A.D.V.Weerasinghe

b. 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.

- Profit=revenue-cost
- Cost=attendance x 3 + 500
- Revenue = attendance x ticket Price
- Profit=(attendance x ticket Price)-(attendance x 3+500)
- **Profit=attendance (ticket Price – 3) + 500**

```scala
@main
def run(){
   prg(5)
   prg(10)
   prg(15)
   prg(20)
}
def prg(tp:Int)= println(profit(revenue(tp),cost(tp)))
def profit(revenue:Int,cost:Int)=revenue-cost
def cost(tp:Int):Int={500+3*attend(tp)}
def revenue(tp:Int)=attend(tp)*tp
def attend(tp:Int):Int={
    if(tp==15)
        120
    else if(tp>15)
        120+((tp-15)/5)*20
    else if(tp<15)
        120-((15-tp)/5)*20
    else
        0
}
```