

## ASSIGNMENT 13.3

### **Problem Statement:**

Find square root of number using Babylonian method.

- 1 Start with an arbitrary positive start value  $x$  (the closer to the root, the better).
- 2 Initialize  $y = 1$ .
3. Do following until desired approximation is achieved.
  - a) Get the next approximation for root using average of  $x$  and  $y$
  - b) Set  $y = n/x$

### **Solution:**

#### **Finding square root of number using Babylonian method:**

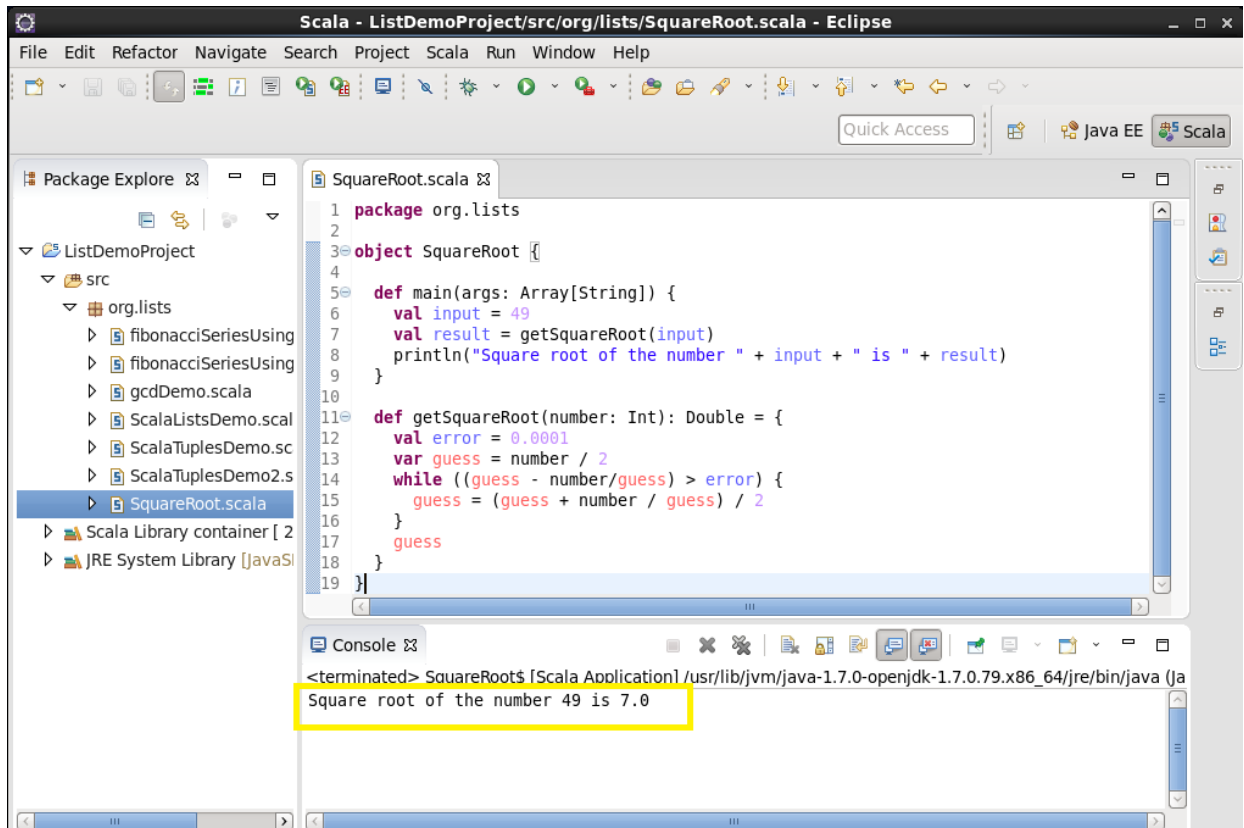
Babylonian method consists of dividing and averaging the numbers with guessed numbers to arrive at the closest value to the square root of given number.

Here is the code snippet I have written in Scala to get this done:

```
object SquareRoot
{
  def main(args: Array[String]) {
    val input = 49
    val result = getSquareRoot(input)           // method invocation
    println("Square root of the number " + input + " is " + result) // print result
  }
  def getSquareRoot(number: Int): Double = {
    val error = 0.0001                         // set an initial value for error
    var guess = number / 2                     // initialize the guess value with number
    while((guess - number/guess) > error) // repeat if difference between guesses is more than error
      guess = (guess + number / guess) / 2    // compute average of guess and the number
    guess                                     // return the guessed number as result
  }
}
```

## Output:

Square root of given number 49 is 7.0



```
Scala - ListDemoProject/src/org/lists/SquareRoot.scala - Eclipse
File Edit Refactor Navigate Search Project Scala Run Window Help
Quick Access Java EE Scala

Package Explorer
ListDemoProject
  src
    org.lists
      fibonacciSeriesUsing
      fibonacciSeriesUsing
      gcdDemo.scala
      ScalaListsDemo.scala
      ScalaTuplesDemo.sc
      ScalaTuplesDemo2.s
      SquareRoot.scala
  Scala Library container [ 2
  JRE System Library [JavaS

SquareRoot.scala
1 package org.lists
2
3 object SquareRoot {
4
5 def main(args: Array[String]) {
6   val input = 49
7   val result = getSquareRoot(input)
8   println("Square root of the number " + input + " is " + result)
9 }
10
11 def getSquareRoot(number: Int): Double = {
12   val error = 0.0001
13   var guess = number / 2
14   while ((guess - number/guess) > error) {
15     guess = (guess + number / guess) / 2
16   }
17   guess
18 }
19 }

Console
<terminated> SquareRoot$ [Scala Application] /usr/lib/jvm/java-1.7.0-openjdk-1.7.0.79.x86_64/jre/bin/java (ja
Square root of the number 49 is 7.0
```

## Steps involved in Babylonian method of finding square root of a number:

Step 1: Make a guess

Step 2: Divide the input number with the guessed number

Step 3: Compute average of these two numbers

Step 4: Use this average as the next guess

Step 5: Repeat the process until the guess gets less than or equal to the value of error.