**Session 13: SCALA - SESSION II**

Assignment 13.3

Student Name: Abarajithan SA

Course: Big Data Hadoop & Spark Training

Start Date:  2017-09-09

End Date:  2017-11-26

**Assignment 13.3**–

Find square root of number using Babylonian method.

Contents

[Introduction 1](#_Toc501107020)

[Problem Statement 2](#_Toc501107021)

[Task – Find square root if a number using Babylonian Method 2](#_Toc501107022)

[Scala code 2](#_Toc501107023)

[Screen Shot 3](#_Toc501107024)

# Introduction

In this assignment, we are going to write a SCALA code to find square root using Babylonian method,

# Problem Statement

Find square root of number using Babylonian method.

1. 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.
   1. Get the next approximation for root using average of x and y
   2. Set y = n/x

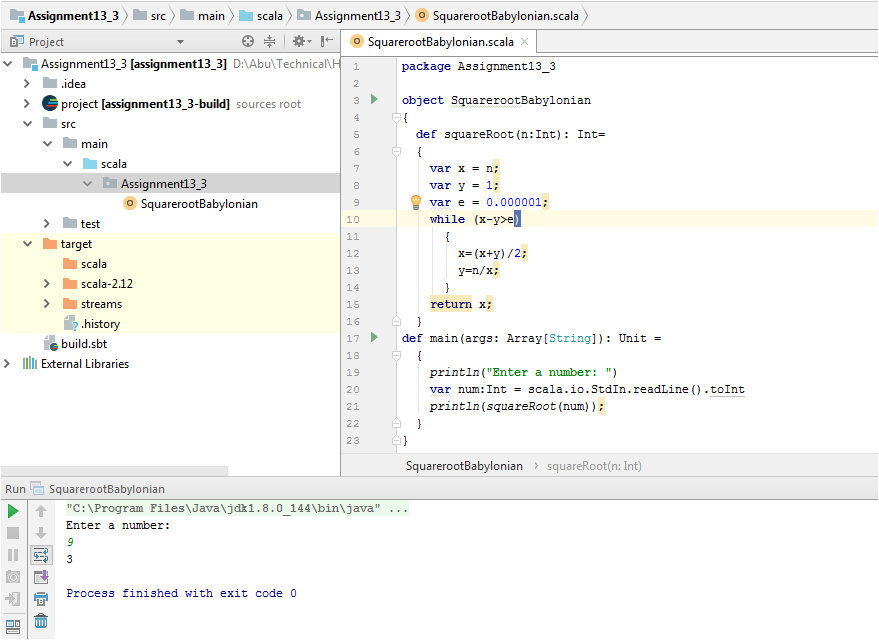
The Babylonian method for finding square roots involves dividing and averaging, over and over, to obtain a more accurate solution with each repeat of the process. Step 2: Divide your original number by your guess. Step 3: Find the average of these numbers. Step 4: Use this average as your next guess.

# Task – Find square root if a number using Babylonian Method

## Scala code

**package** Assignment13\_3  
  
**object** SquarerootBabylonian  
{  
 **def** squareRoot(n:Int): Int=  
 {  
 **var** x = n;  
 **var** y = 1;  
 **var** e = 0.000001;  
 **while** (x-y>e)  
 {  
 x=(x+y)/2;  
 y=n/x;  
 }  
 **return** x;  
 }  
**def** main(args: Array[String]): Unit =  
 {  
 *println*(**"Enter a number: "**)  
 **var** num:Int = scala.io.StdIn.readLine().toInt  
 *println*(*squareRoot*(num));  
 }  
}

## Screen Shot



# Output

If we enter a number 64, the square root of that value is 8

