Session 13 : SCALA Session II

Assignment 3 Questions

**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.
4. Get the next approximation for root using average of x and y
5. Set y = n/x

**Solution:-**

Scala Project - 

Code –

/\*\*

\* Author Nihit Rai

\* This class provides method to find square root of a number

\* using babylonian method

\*/

**class** BabylonianSqRoot {

/\*\*

\* Method to return square root of number passed as parameter

\*/

**def** findSqRt(n: Double): Double = {

**var** root: Double = 0.0;

//Starting with an arbitrary positive start value x (the closer to the root, the better)

//setting x = n

**var** x = n;

//Initialize y = 1.

**var** y:Double = 1;

//e decides the accuracy level

**var** e: Double = 0.000001;

//Do following until desired approximation is achieved

**while**(x - y > e)

{

//Get the next approximation for root using average of x and y

x = (x + y)/2;

//Set y = n/x

y = n/x;

}

root = x;

//Rounding off the result to six decimal places

**return** BigDecimal(root).setScale(6, BigDecimal.RoundingMode.HALF\_UP).toDouble;

}

}

//Main object

**object** mainObj

{

**def** main(args: Array[String]) {

**var** obj = **new** BabylonianSqRoot();

**val** number: Double = 9;

//Passing number as parameter to BabylonianSqRoot.findSqRt method

//to determine square root of number using babylonian method

println(" Square root of "+number+" is: "

+obj.findSqRt(number));

}

}

Output Screenshots:-





