**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:

**package** com.assignment.babylonian

//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

**object** SquareRoot {

**def** calcSquareRoot(n: Int): Double = {

**var** x = n.toDouble

**var** y = 1.0

**var** e = 0.0001

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

x = (x + y) / 2;

y = n / x;

}

**return** x;

}

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

**var** n = 2

print("Square Root a of " + n + " is :\t" + calcSquareRoot(n))

println()

n = 4;

print("Square Root a of " + n + " is :\t" + calcSquareRoot(n))

n = 14;

print("Square Root a of " + n + " is :\t" + calcSquareRoot(n))

}

}