SESSION 13 ASSIGNMENT 3

**Problem**:

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

**Code**:

package Assignment\_13

class session\_13\_3(a:Int) {

//Initialize variable x

var x:Float = (a.toString.length.toFloat)

var y:Float = 1

var temp:Float = 1

if (a == 0)

println("please input whole number")

else

{

sqrt(x)

println("Square Root of " + a + " to nearest integer is " + y.toInt )

}

*//c is the number whose squareroot we need*

def sqrt (c:Float): Unit = {

temp = y

y = (c + (a/c))/2

if (temp != y) sqrt(y)

}

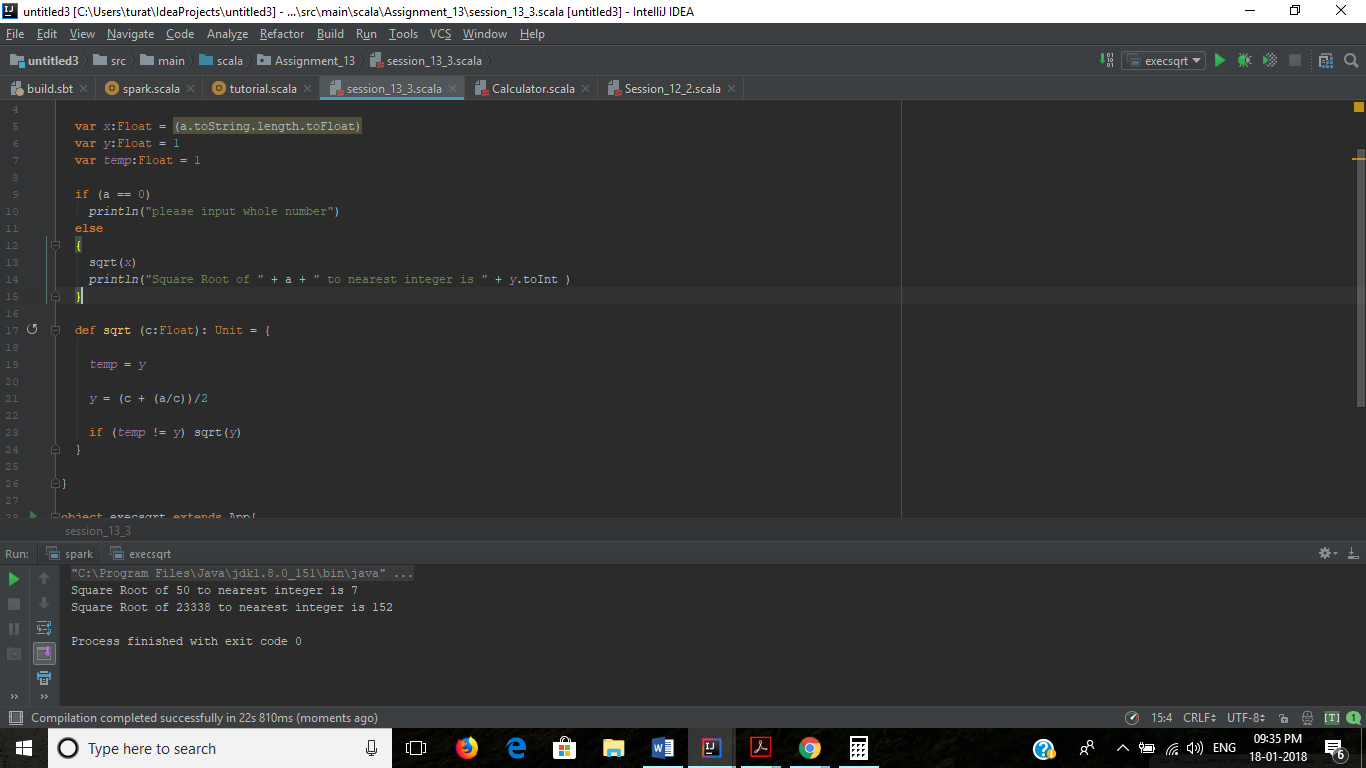
}

object execsqrt extends App{

var sqrt = new session\_13\_3(50)

var sqrt1 = new session\_13\_3(23338)

}



**Output**:

Square Root of 50 to nearest integer is 7

Square Root of 23338 to nearest integer is 152

Process finished with exit code 0