

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

Solution:-

Scala Application for finding square root of number using Babylonian Method is as follows:-

```
object square_root {  
  
  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]) {
```

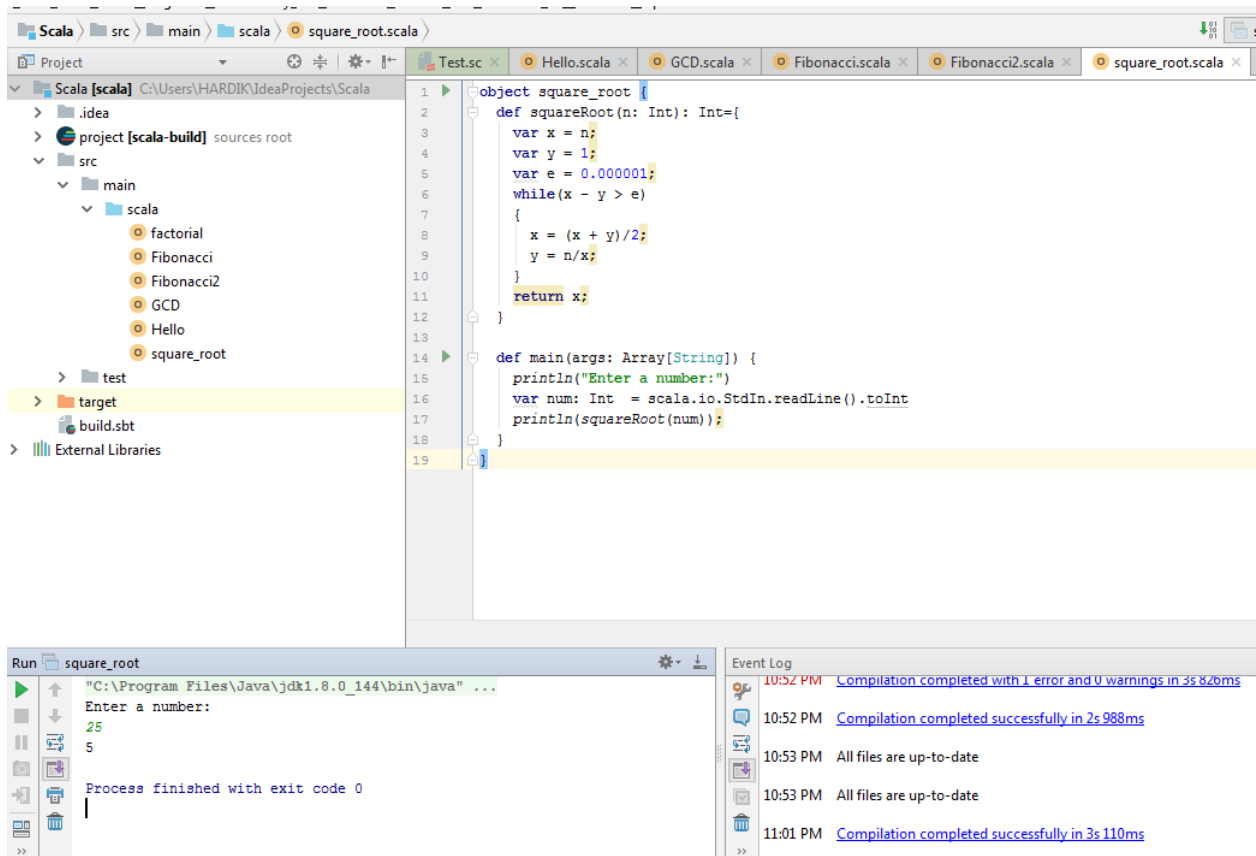
```
    println("Enter a number:")
```

```
    var num: Int = scala.io.StdIn.readLine().toInt
```

```
    println(squareRoot(num));
```

```
}
```

```
}
```



Scala [scala] C:\Users\HARDIK\IdeaProjects\Scala

Project: Scala [scala] C:\Users\HARDIK\IdeaProjects\Scala

- src
 - main
 - scala
 - factorial
 - Fibonacci
 - Fibonacci2
 - GCD
 - Hello
 - square_root
- target
- build.sbt
- External Libraries

```
1 object square_root {  
2   def squareRoot(n: Int): Int={  
3     var x = n;  
4     var y = 1;  
5     var e = 0.000001;  
6     while(x - y > e)  
7     {  
8       x = (x + y)/2;  
9       y = n/x;  
10    }  
11    return x;  
12  }  
13  
14  def main(args: Array[String]) {  
15    println("Enter a number:")  
16    var num: Int = scala.io.StdIn.readLine().toInt  
17    println(squareRoot(num));  
18  }  
19 }
```

Run square_root

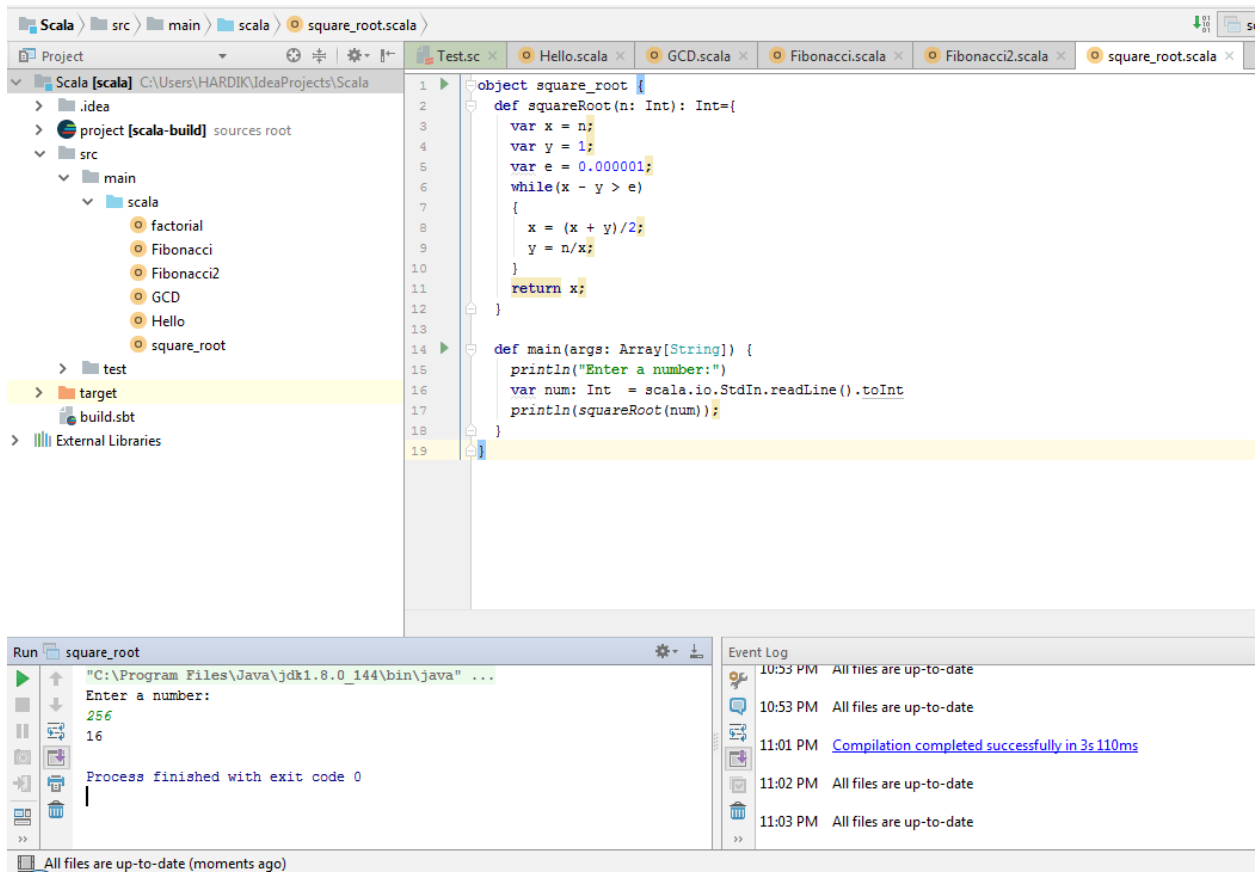
"C:\Program Files\Java\jdk1.8.0_144\bin\java" ...

Enter a number:
1
1

Process finished with exit code 0

Event Log

- 10:52 PM [Compilation completed successfully in 25.988ms](#)
- 10:53 PM All files are up-to-date
- 10:53 PM All files are up-to-date
- 11:01 PM [Compilation completed successfully in 3s.110ms](#)
- 11:02 PM All files are up-to-date



Submitted By:-

Hardik Kaushik