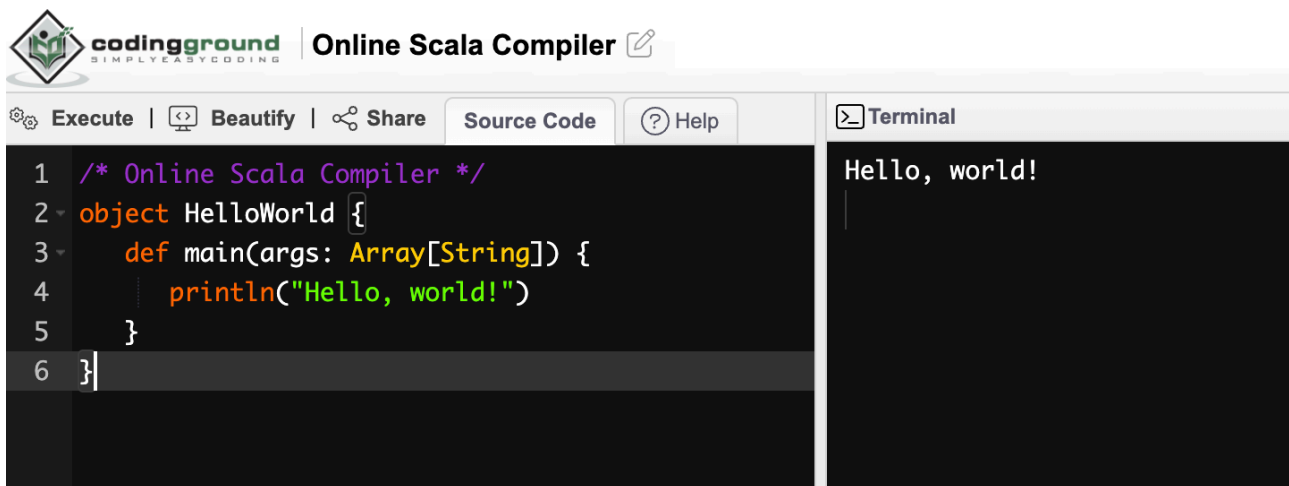


BDA LAB
SCALA PROGS
Mallika Prasad
1BM19CS081

9. Program to print word count on scala shell and print “Hello world” on scala IDE/* Online Scala Compiler */

```
object HelloWorld {  
  def main(args: Array[String]) {  
    println("Hello, world!")  
  }  
}
```



Word Count-

```
scala> val rdd3 = sc.textFile("/Users/mallikaprasad/Desktop/sample1.txt")  
rdd3: org.apache.spark.rdd.RDD[String] = /Users/mallikaprasad/Desktop/sample1.txt  
MapPartitionsRDD[7] at textFile at <console>:23  
  
scala> rdd3.collect();  
res5: Array[String] = Array("hello how are hello ", "how ")  
  
scala> val split= rdd3.flatMap(line=>line.split(" "))  
split: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[8] at flatMap at <console>:23  
  
scala> split.collect();  
res6: Array[String] = Array(hello, how, are, hello, how)  
  
scala> val mapped= split.map(w=>(w,1))  
mapped: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[9] at map at  
<console>:23
```

```
scala> mapped.collect()
res7: Array[(String, Int)] = Array((hello,1), (how,1), (are,1), (hello,1), (how,1))

scala> val red= mapped.reduceByKey(_+_ )
red: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[10] at reduceByKey at
<console>:23

scala> red.collect()
res8: Array[(String, Int)] = Array((are,1), (how,2), (hello,2))
```

```
scala> val rdd3 = sc.textFile("/Users/mallikaprasad/Desktop/sample1.txt")
rdd3: org.apache.spark.rdd.RDD[String] = /Users/mallikaprasad/Desktop/sample1.txt MapPartitionsRDD[7] at textFile at <console>:23

scala> rdd3.collect();
res5: Array[String] = Array("hello how are hello ", "how ")

scala> val split= rdd3.flatMap(line=>line.split(" "))
split: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[8] at flatMap at <console>:23

scala> split.collect();
res6: Array[String] = Array(hello, how, are, hello, how)

scala> val mapped= split.map(w=>(w,1))
mapped: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[9] at map at <console>:23

scala> mapped.collect()
res7: Array[(String, Int)] = Array((hello,1), (how,1), (are,1), (hello,1), (how,1))

scala> val red= mapped.reduceByKey(_+_ )
red: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[10] at reduceByKey at <console>:23

scala> red.collect()
res8: Array[(String, Int)] = Array((are,1), (how,2), (hello,2))
```

10. Using RDD and FlatMap count how many times each word appears in a file and write out a list of words whose count is strictly greater than 4 using Spark

```
Scala> val ip=sc.textFile("/Users/mallikaprasad/Desktop/sample1.txt");
ip: org.apache.spark.rdd.RDD[String] = /Users/mallikaprasad/Desktop/sample1.txt
MapPartitionsRDD[38] at textFile at <console>:24

scala> ip.collect();
res31: Array[String] = Array(hello hello how are how How hello hello hi are hello)

scala> val split=ip.flatMap(line=>line.split(" "))
split: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[39] at flatMap at <console>:24

scala> split.collect();
res32: Array[String] = Array(hello, hello, how, are, how, How, hello, hello, hi, are, hello)

scala> val mapped= split.map(w=>(w,1))
mapped: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[40] at map at
<console>:24
```

```
scala> mapped.collect();
res33: Array[(String, Int)] = Array((hello,1), (hello,1), (how,1), (are,1), (how,1), (How,1),
(hello,1), (hello,1), (hi,1), (are,1), (hello,1))
```

```
scala> val red= mapped.reduceByKey(_+_ )
red: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[41] at reduceByKey at
<console>:24
```

```
scala> red.collect()
res34: Array[(String, Int)] = Array((are,2), (how,2), (hello,5), (How,1), (hi,1))
```

```
scala> scala> val fil= red.filter(f=>f._2>4)
```

```
scala> val fil= red.filter(f=>f._2>4)
fil: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[42] at filter at <console>:24
```

```
scala> fil.collect();
res35: Array[(String, Int)] = Array((hello,5))
```

```
scala> val split=ip.flatMap(line=>line.split(" "))
split: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[39] at flatMap at <console>:24

scala> split.collect();
res32: Array[String] = Array(hello, hello, how, are, how, How, hello, hello, hi, are, hello)

scala> val mapped= split.map(w=>(w,1))
mapped: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[40] at map at <console>:24

scala> mapped.collect();
res33: Array[(String, Int)] = Array((hello,1), (hello,1), (how,1), (are,1), (how,1), (How,1), (hello,1), (hello,1), (hi,1),
(are,1), (hello,1))

scala> val red= mapped.reduceByKey(_+_ )
red: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[41] at reduceByKey at <console>:24

scala> red.collect()
res34: Array[(String, Int)] = Array((are,2), (how,2), (hello,5), (How,1), (hi,1))

scala> scala> val fil= red.filter(f=>f._2>4)

// Detected repl transcript. Paste more, or ctrl-D to finish.
\

// Replaying 1 commands from transcript.

scala> val fil= red.filter(f=>f._2>4)
fil: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[42] at filter at <console>:24

scala> fil.collect();
res35: Array[(String, Int)] = Array((hello,5))
```