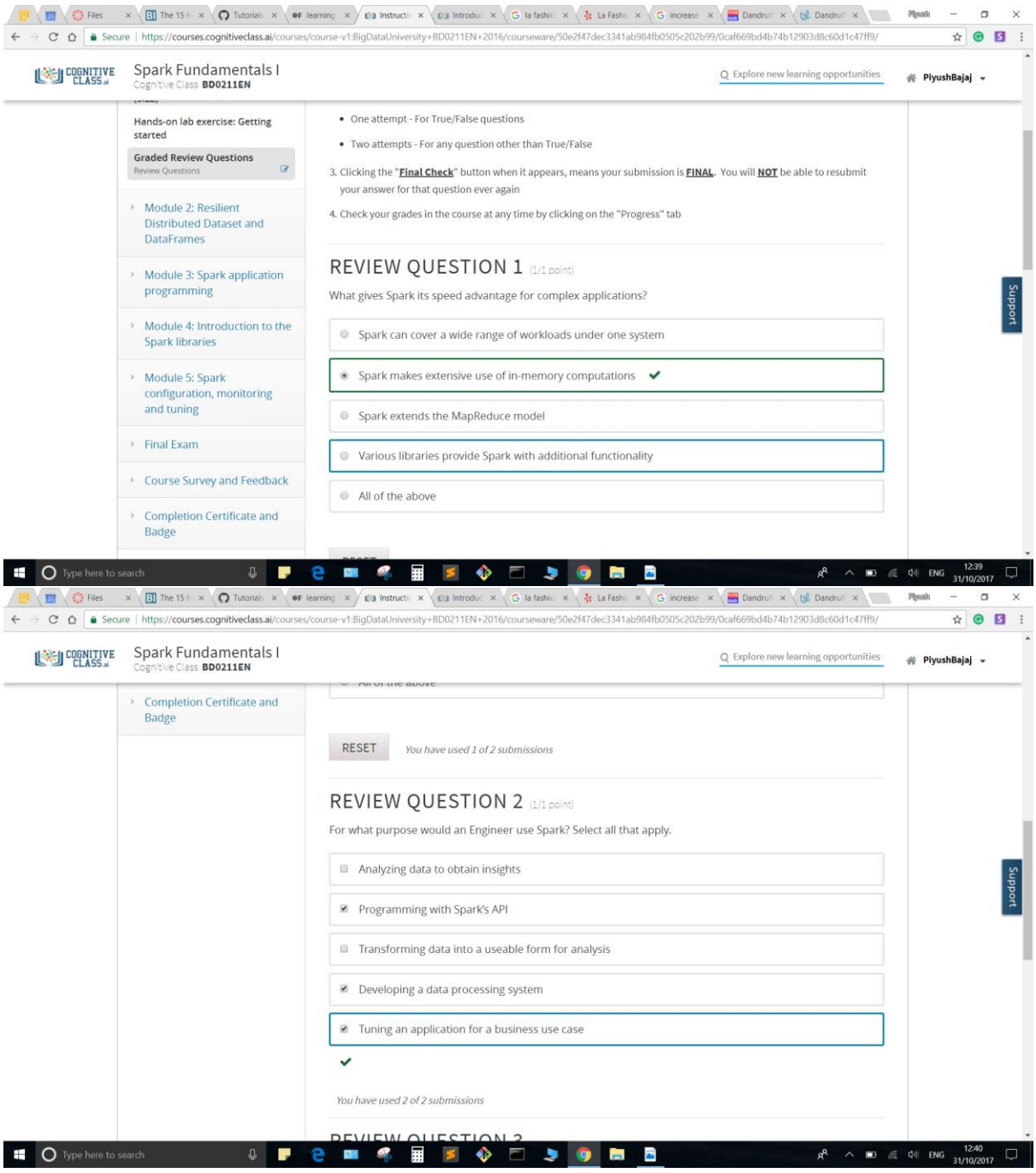
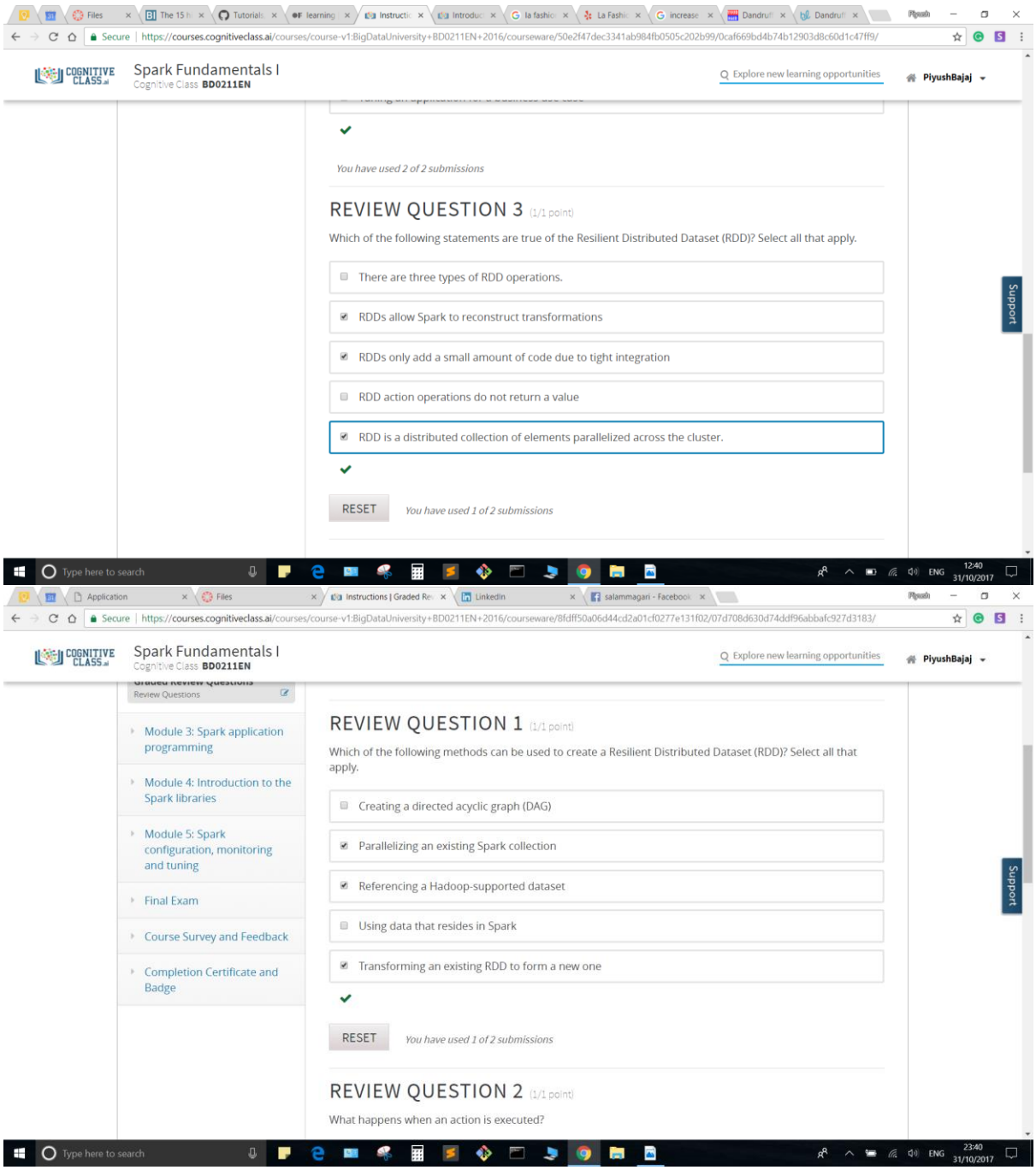
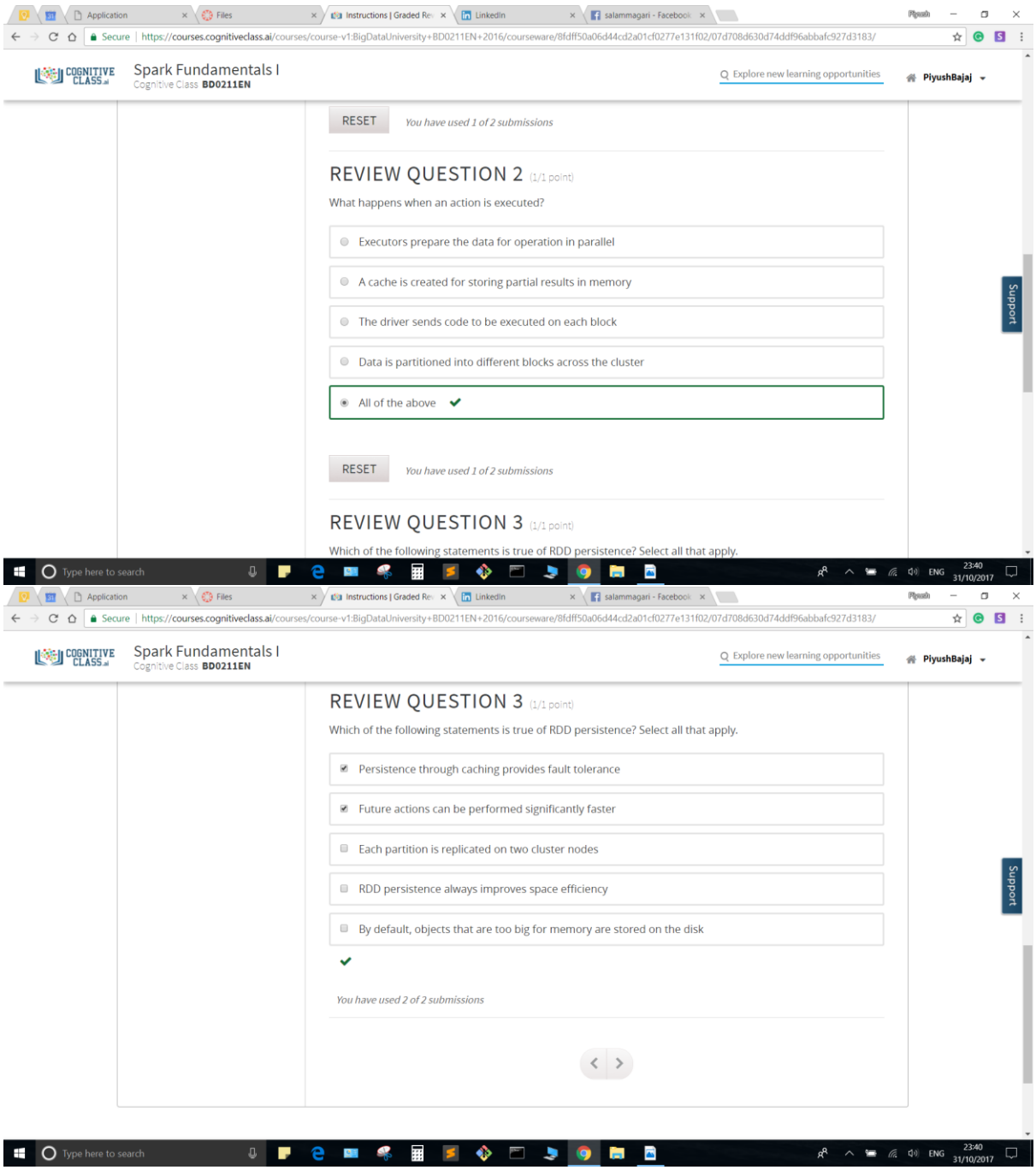
Lab #8 – Spark

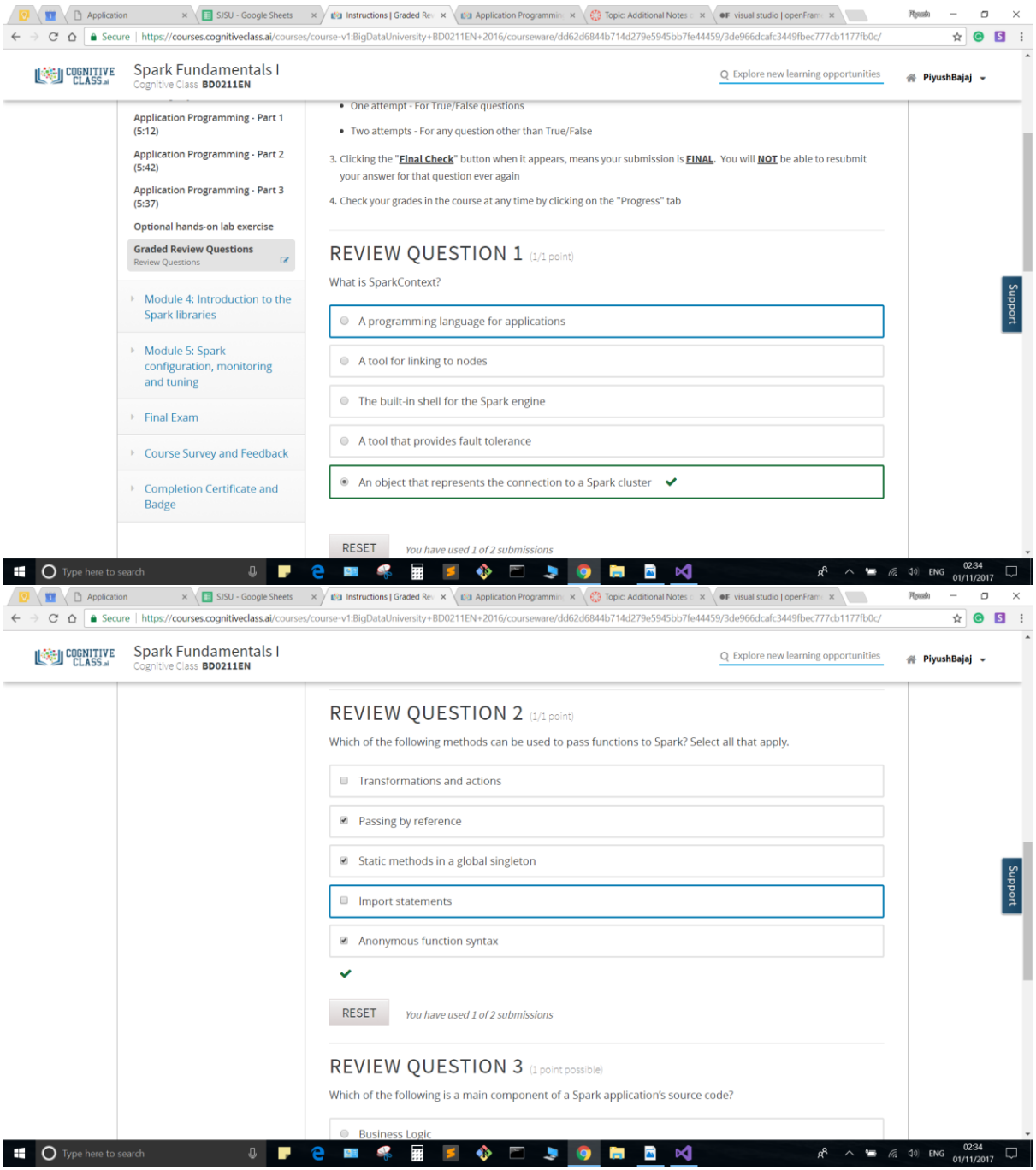
1. Answer the Review Questions for Module 1,2 and 3. Take a screenshot of both answered reviews.

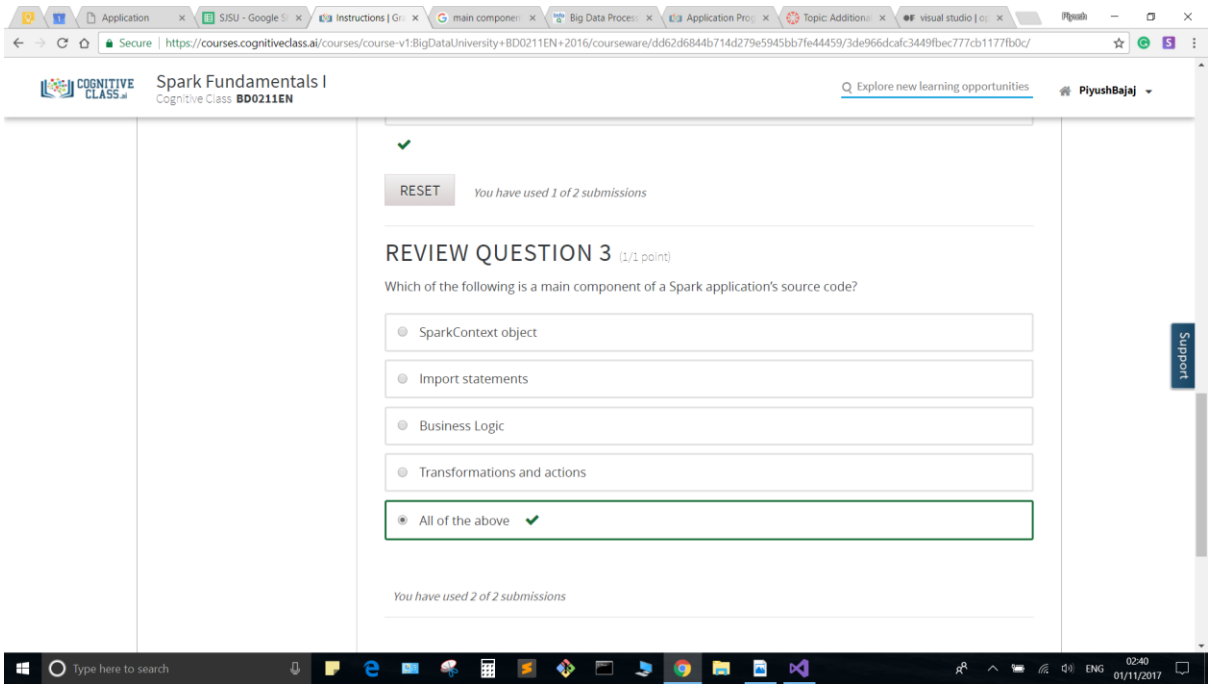



Spark Fundamentals I
Cognitive Class BD0211EN
Explore new learning opportunities
PiyushBajaj
You have used 2 of 2 submissions
REVIEW QUESTION 3 (1/1 point)
Which of the following statements are true of the Resilient Distributed Dataset (RDD)? Select all that apply.
There are three types of RDD operations.
RDDs allow Spark to reconstruct transformations
RDDs only add a small amount of code due to tight integration
RDD action operations do not return a value
RDD is a distributed collection of elements parallelized across the cluster.
RESET
You have used 1 of 2 submissions
Support
Spark Fundamentals I
Cognitive Class BD0211EN
Explore new learning opportunities
PiyushBajaj
Review Questions
Module 3: Spark application programming
Module 4: Introduction to the Spark libraries
Module 5: Spark configuration, monitoring and tuning
Final Exam
Course Survey and Feedback
Completion Certificate and Badge
REVIEW QUESTION 1 (1/1 point)
Which of the following methods can be used to create a Resilient Distributed Dataset (RDD)? Select all that apply.
Creating a directed acyclic graph (DAG)
Parallelizing an existing Spark collection
Referencing a Hadoop-supported dataset
Using data that resides in Spark
Transforming an existing RDD to form a new one
RESET
You have used 1 of 2 submissions
REVIEW QUESTION 2 (1/1 point)
What happens when an action is executed?
Support

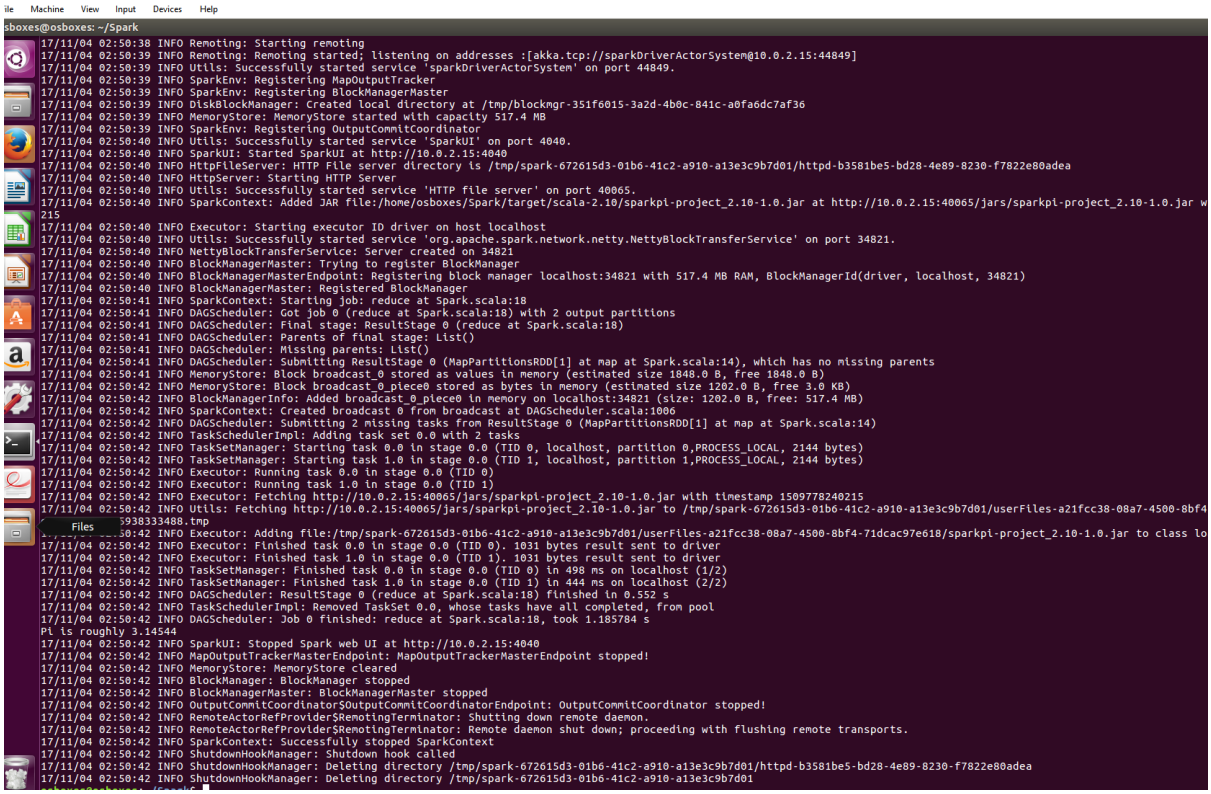


Submitted by Piyush Bajaj

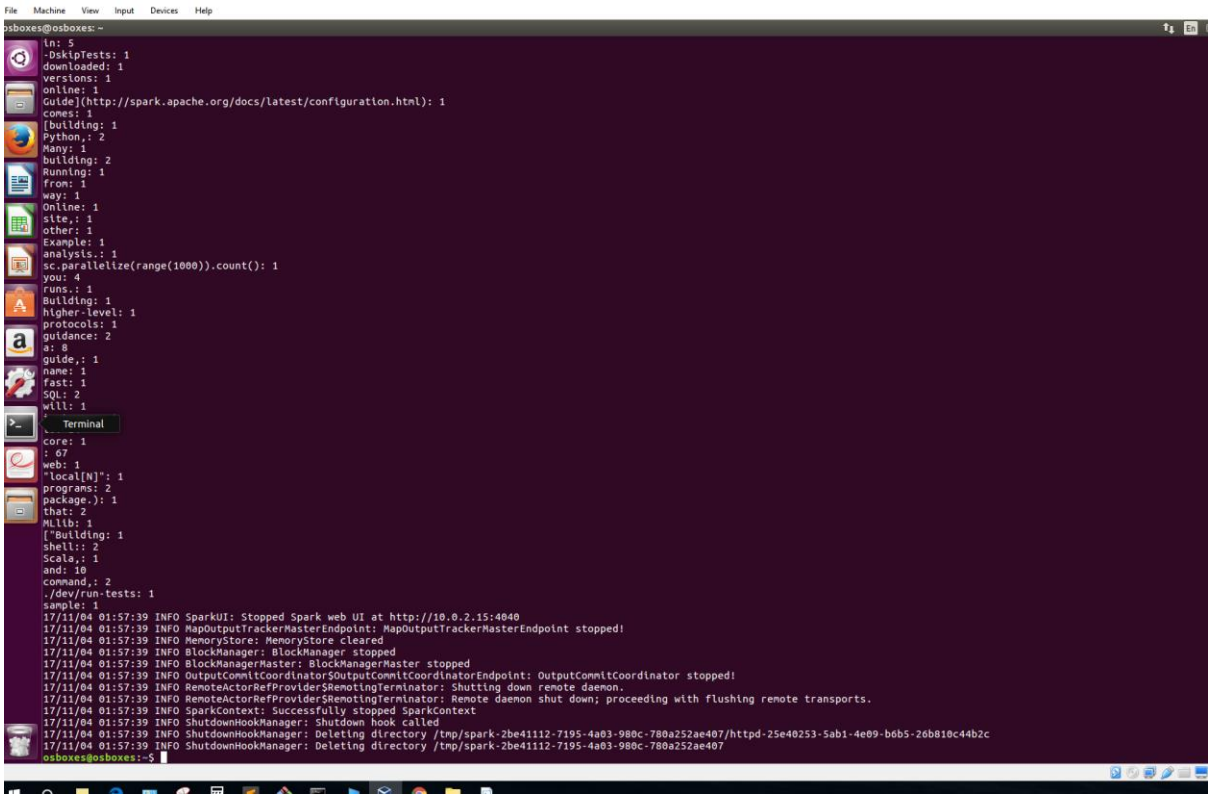

Spark Fundamentals I
Cognitive Class BD0211EN
Explore new learning opportunities
PiyushBajaj
Application Programming - Part 1 (5:12)
Application Programming - Part 2 (5:42)
Application Programming - Part 3 (5:37)
Optional hands-on lab exercise
Graded Review Questions
Review Questions
Module 4: Introduction to the Spark libraries
Module 5: Spark configuration, monitoring and tuning
Final Exam
Course Survey and Feedback
Completion Certificate and Badge
One attempt - For True/False questions
Two attempts - For any question other than True/False
3. Clicking the "Final Check" button when it appears, means your submission is FINAL. You will NOT be able to resubmit your answer for that question ever again
4. Check your grades in the course at any time by clicking on the "Progress" tab
REVIEW QUESTION 1 (1/1 point)
What is SparkContext?
A programming language for applications
A tool for linking to nodes
The built-in shell for the Spark engine
A tool that provides fault tolerance
An object that represents the connection to a Spark cluster
RESET
You have used 1 of 2 submissions
Support
REVIEW QUESTION 2 (1/1 point)
Which of the following methods can be used to pass functions to Spark? Select all that apply.
Transformations and actions
Passing by reference
Static methods in a global singleton
Import statements
Anonymous function syntax
RESET
You have used 1 of 2 submissions
REVIEW QUESTION 3 (1 point possible)
Which of the following is a main component of a Spark application's source code?
Business Logic



2. b. Run the Sample Scala application to calculate the value of Pi. Take a screenshot of the output of the run of spark-submit.



3. Spark can be run with Java or Python as well. Continuing the same hands-on lab exercise, setup/write the WordCount program in Java based on the sample given. Run the wordcount program and take a screenshot of the output. Try the program with your own input textfile afterwards.



Full output:

package: 1
For: 2
Programs: 1
processing.: 1
Because: 1
The: 1
cluster.: 1
its: 1
[run: 1
APIs: 1
have: 1
Try: 1
computation: 1
through: 1
several: 1
This: 2
graph: 1
Hive: 2
storage: 1
["Specifying: 1
To: 2

Submitted by Piyush Bajaj

page](http://spark.apache.org/documentation.html): 1
Once: 1
"yarn": 1
prefer: 1
SparkPi: 2
engine: 1
version: 1
file: 1
documentation,: 1
processing,: 1
the: 21
are: 1
systems.: 1
params: 1
not: 1
different: 1
refer: 2
Interactive: 2
R,: 1
given.: 1
17/11/04 01:57:39 INFO TaskSetManager: Finished task 0.0 in stage 1.0 (TID 1) in 192 ms on localhost (1/1)
if: 4
build: 3
17/11/04 01:57:39 INFO TaskSchedulerImpl: Removed TaskSet 1.0, whose tasks have all completed, from pool
when: 1
be: 2
Tests: 1
Apache: 1
./bin/run-example: 2
programs,: 1
including: 3
Spark.: 1
package.: 1
1000).count(): 1
Versions: 1
HDFS: 1
Data.: 1
>>>: 1
programming: 1
Testing: 1
module,: 1
Streaming: 1
environment: 1
run:: 1
clean: 1
1000:: 2
rich: 1
GraphX: 1
Please: 3
is: 6
run: 7
URL,: 1
threads.: 1
same: 1
MASTER=spark://host:7077: 1
on: 5
built: 1
against: 1

[Apache: 1
tests: 2
examples: 2
at: 2
optimized: 1
usage: 1
using: 2
graphs: 1
talk: 1
Shell: 2
class: 2
abbreviated: 1
directory.: 1
README: 1
computing: 1
overview: 1
`examples`: 2
example:: 1
##: 8
N: 1
set: 2
use: 3
Hadoop-supported: 1
tests](https://cwiki.apache.org/confluence/display/SPARK/Useful+Developer+Tools).: 1
running: 1
find: 1
contains: 1
project: 1
Pi: 1
need: 1
or: 3
Big: 1
Java,: 1
high-level: 1
uses: 1
<class>: 1
Hadoop,: 2
available: 1
requires: 1
(You: 1
see: 1
Documentation: 1
of: 5
tools: 1
using:: 1
cluster: 2
must: 1
supports: 2
built,: 1
system: 1
build/mvn: 1
Hadoop: 3
this: 1
Version"](http://spark.apache.org/docs/latest/building-spark.html#specifying-the-hadoop-version): 1
particular: 2
Python: 2
Spark: 13
general: 2

YARN,: 1
pre-built: 1
[Configuration: 1
locally: 2
library: 1
A: 1
locally.: 1
sc.parallelize(1: 1
only: 1
Configuration: 1
following: 2
basic: 1
#: 1
changed: 1
More: 1
which: 2
learning,: 1
first: 1
./bin/pyspark: 1
also: 4
should: 2
for: 11
[params]`.: 1
documentation: 3
[project: 2
mesos://: 1
Maven](http://maven.apache.org/).: 1
setup: 1
<http://spark.apache.org/>: 1
latest: 1
your: 1
MASTER: 1
example: 3
scala>: 1
DataFrames,: 1
provides: 1
configure: 1
distributions.: 1
can: 6
About: 1
instructions.: 1
do: 2
easiest: 1
no: 1
how: 2
`./bin/run-example: 1
Note: 1
individual: 1
spark://: 1
It: 2
Scala: 2
Alternatively,: 1
an: 3
variable: 1
submit: 1
machine: 1
thread,: 1
them,: 1

detailed: 2
stream: 1
And: 1
distribution: 1
return: 2
Thriftserver: 1
./bin/spark-shell: 1
"local": 1
start: 1
You: 3
Spark](#building-spark).: 1
one: 2
help: 1
with: 3
print: 1
Spark"](http://spark.apache.org/docs/latest/building-spark.html).: 1
data: 1
wiki](https://cwiki.apache.org/confluence/display/SPARK).: 1
in: 5
-DskipTests: 1
downloaded: 1
versions: 1
online: 1
Guide](http://spark.apache.org/docs/latest/configuration.html): 1
comes: 1
[building: 1
Python,: 2
Many: 1
building: 2
Running: 1
from: 1
way: 1
Online: 1
site,: 1
other: 1
Example: 1
analysis.: 1
sc.parallelize(range(1000)).count(): 1
you: 4
runs.: 1
Building: 1
higher-level: 1
protocols: 1
guidance: 2
a: 8
guide,: 1
name: 1
fast: 1
SQL: 2
will: 1
instance:: 1
to: 14
core: 1
: 67
web: 1
"local[N]": 1
programs: 2
package.): 1

that: 2
MLlib: 1
["Building: 1
shell:: 2
Scala,: 1
and: 10
command,: 2
./dev/run-tests: 1
sample: 1

Extra Credit:

5b. b. Create your own input text file. Run the wordcount program. Paste the contents of your input textfile along with the contents of parts-00000 and part-00001.