Logic for the program:

1. take the arguments - brokerid, groupid and topics
2. Set a unique name for the application
3. Create context with batch interval
4. set log level to WARN
5. Define a new Hashmap for holding the Kafka information
6. Set KafkaParams
7. Start streaming
8. Create direct kafka stream with brokers and topics and take JavaInputDStream
9. Map the incoming JSON from kafka stream to the object.
10. Iterate through the processed objects
11. Filter ETH, LTC, XRP and BTC currencies
12. Use mapToPair to map <key (currency name), closing#opening#volume>
13. Use mapValues to get the count as -- <key, <closing#opening#volume, count>>
14. Problem 1 - Iterate through the RDD calulate avaerage closing
15. Problem 2 - Iterate through the RDD calulate max profit
16. Problem 2 - Iterate through the RDD calulate max volume
17. Stop the streaming
18. Terminate

Command to run in eclipse ->

1. select run as configuration
2. Pass arguments : broker - 34.206.133.62:9092, groupid (any random text) and topics - "stockData"

Command to run on ec2:

```
cd <full-path-for-fat-jar>
java -jar tutorial.KafkaConsumerDriver 34.206.133.62:9092 <groupid> stockData
```