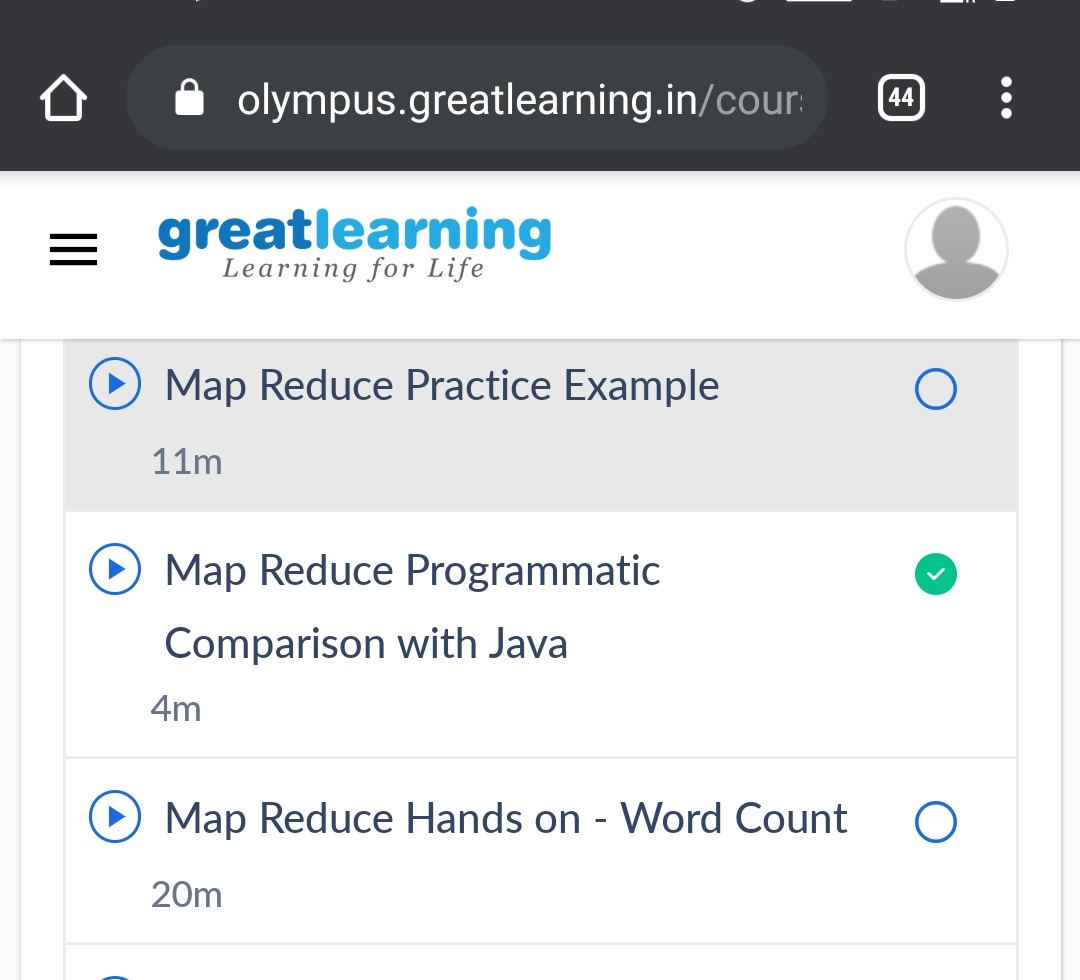
**DAILY ONLINE ACTIVITIES SUMMARY**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Date:** | **26/05/2020** | | | | **Name:** | **Samrin banu** | |
| **Sem & Sec** | **8th B** | | | | **USN:** | **4AL16Cs082** | |
| Online Test Summary | | | | | | | |
| **Subject** | | **BDA** | | | | | |
| **Max. Marks** | | **40** | | **Score** | | **30** | |
| Certification Course Summary | | | | | | | |
| **Course** | **Introduction to Hadoop** | | | | | | |
| **Certificate Provider** | | | **Great learning** | **Duration** | | | **20 mins** |
| Coding Challenges | | | | | | | |
| **Problem Statement:**  **1) write a python program to read a numbernand print the series"1+2.."** | | | | | | | |
| **Status: Solved** | | | | | | | |
| **Uploaded the report in Github** | | | | **YES** | | | |
| **If yes Repository name** | | | | **Samrinbanu** | | | |
| **Uploaded the report in slack** | | | | **YES** | | | |

**Certification Course Details:**



# Map Reduce Programmatic Comparison with Java:

# MapReduce is a framework that programmers today can use to write applications that are able to process massive quantities of data using a modern distributed data processing approach. This processing approach is very popular among organizations today. As it allows the processing of data in parallel on large commodity hardware clusters, using MapReduce can significantly speed up data processing.

# In this video, we will take a look at how to execute MapReduce over data stored in Redis using Redisson - Redis based In-Memory Data Grid for Java.

# Redisson is a state-of-the-art Redis client that opens up near infinite possibilities for programming and data processing using Java. A wide range of companies, from the biggest enterprises to the smallest startups, use Redisson to empower their Java applications with Redis.

# CODE:

Program no:1

###**write a python program to read a numbernand print the series"1+2..**###

n=int(input("EnteraNumber:"))

series\_sum=[]

foriinrange(1,n+1):

series\_sum.append(i)

ifi==n:

print(i,end='')

else:

print(i,end='+')

print('=',sum(series\_sum),sep='')