

DAILY ONLINE ACTIVITIES SUMMARY

Date:	03-06-2020	Name:	D Jasmine Joyline
Sem & Sec	VI Sem A	USN:	4AL17CS024
Online Test Summary			
Subject	-		
Max. Marks	-	Score	-
Certification Course Summary			
Course	Python Bootcamp 2020:Build 15 working applications and games		
Certificate Provider	Udemy	Duration	32hr
Coding Challenges			
Problem Statement: <ol style="list-style-type: none"> 1. Write a python program to generate prime number in an interval 2. Write a Java Program to Implement Circular Doubly Linked List 			
Status:Completed			
Uploaded the report in Github		Yes	
If yes Repository name		https://github.com/alvas-education-foundation/D_Jasmine_Joyline/tree/master/daily_progress	
Uploaded the report in slack		Yes	

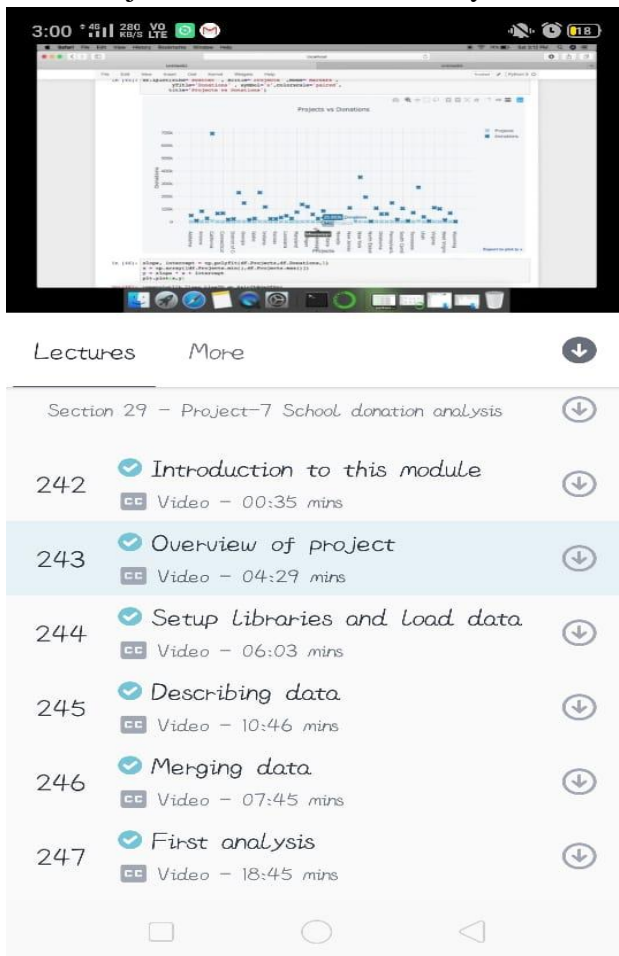
Online Test Details:

Not conducted

Certification Course Details:

Module that I have completed today:

- Database
- Project7:School donation analysis

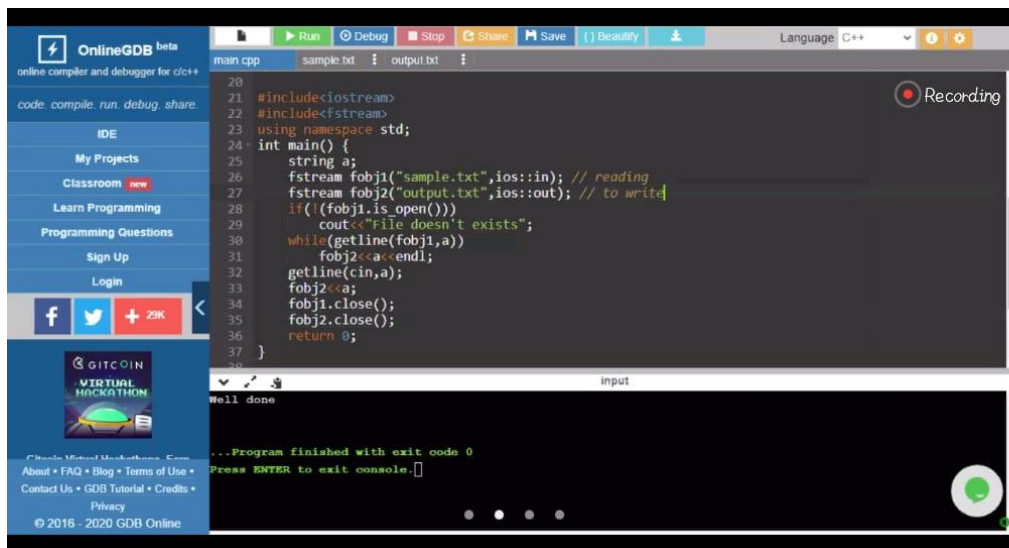


The screenshot shows a mobile application interface. At the top, there's a status bar with the time 3:00, signal strength, 250 KB/S LTE, and battery level at 18%. Below the status bar is a video player displaying a scatter plot titled 'Projects vs Donations'. The plot has 'Projects' on the x-axis and 'Donations' on the y-axis, with blue data points. Below the video player, there are two tabs: 'Lectures' and 'More'. The 'Lectures' tab is active, showing a list of video lectures. The list starts with 'Section 29 - Project-7 School donation analysis' and includes several individual lectures, each with a checkmark icon, a title, and a duration. The lecture 'Overview of project' (243) is currently selected and highlighted in blue.

Section	Title	Duration
Section 29	Project-7 School donation analysis	
242	Introduction to this module	00:35 mins
243	Overview of project	04:29 mins
244	Setup Libraries and Load data	06:03 mins
245	Describing data	10:46 mins
246	Merging data	07:45 mins
247	First analysis	18:45 mins

Online C++ classes

- File handling
- Enumerations
- Preprocessor
- CommandLine Arguments



Coding Challenges Details:

1. Write a python program to generate prime number in an interval

You just need to take two number as input from stdin and you need to find prime numbers between those two numbers and print them.

Input Format

You will be taking two numbers as an input from stdin one on each line respectively.

Constraints

$1 \leq A, B \leq 10^4$

Output Format

You need to print the prime numbers one on each line

sample example:

Input

900

1000

Output

907

911

919

929

937

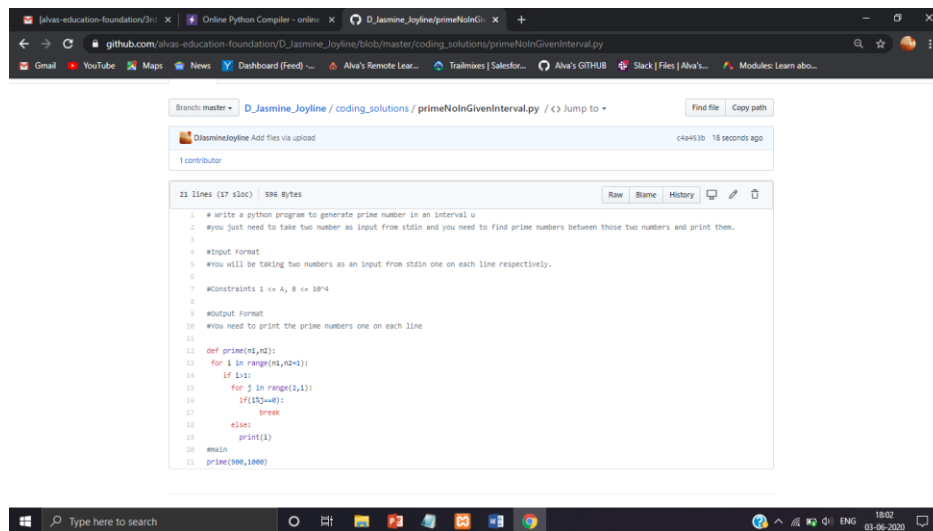
941

947

953

967

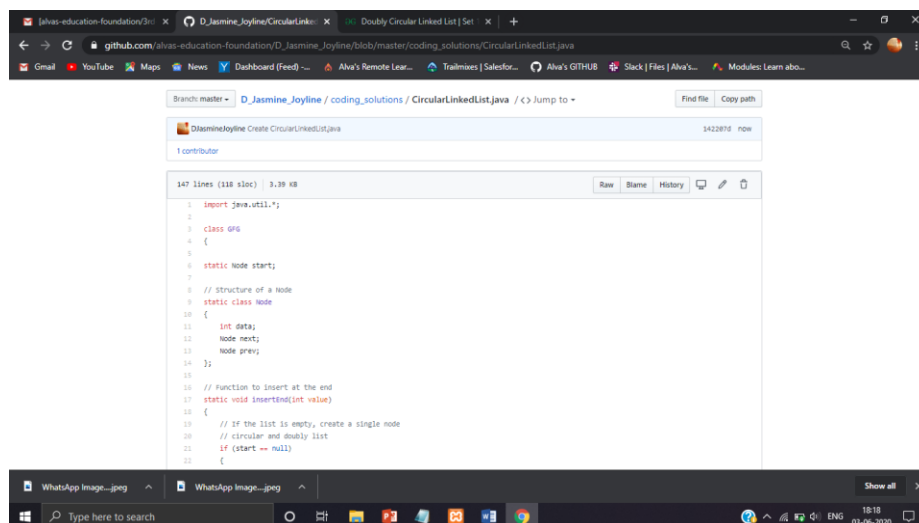
971
977
983
991
997



The screenshot shows a web browser displaying a GitHub repository page for a Python program. The repository is named 'D_Jasmine_Joyline / coding_solutions / primeNoInGivenInterval.py'. The file is 21 lines long (17 sloc) and 596 Bytes. The code is a Python program that takes two numbers as input from stdin and prints the prime numbers between them. The code is as follows:

```
1 # write a python program to generate prime number in an interval u
2 #you just need to take two number as input from stdin and you need to find prime numbers between those two numbers and print them.
3
4 #input format
5 #you will be taking two numbers as an input from stdin one on each line respectively.
6
7 #constraints 1 <= A, B <= 10^4
8
9 #output format
10 #you need to print the prime numbers one on each line
11
12 def prime(n1,n2):
13     for i in range(n1,n2+1):
14         if i<1:
15             for j in range(i,1):
16                 if(i%j==0):
17                     break
18             else:
19                 print(i)
20
21 #main
22 prime(989,1000)
```

2. Write a Java Program to Implement Circular Doubly Linked List



The screenshot shows a web browser displaying a GitHub repository page for a Java program. The repository is named 'D_Jasmine_Joyline / coding_solutions / CircularLinkedList.java'. The file is 147 lines long (118 sloc) and 3.39 KB. The code is a Java program that implements a circular doubly linked list. The code is as follows:

```
1 import java.util.*;
2
3 class GFG
4 {
5
6     static Node start;
7
8     // structure of a Node
9     static class Node
10     {
11         int data;
12         Node next;
13         Node prev;
14     };
15
16     // Function to insert at the end
17     static void insertEnd(int value)
18     {
19         // If the list is empty, create a single node
20         // circular and doubly list
21         if (start == null)
22         {
23             Node n = new Node(value);
24             n.next = n;
25             n.prev = n;
26             start = n;
27         }
28         else
29         {
30             Node n = new Node(value);
31             n.next = start.next;
32             n.prev = start;
33             start.next = n;
34             start = n;
35         }
36     }
37 }
```