DAILY ONLINE ACTIVITIES SUMMARY

Date:	**16-07-2020**	**Name:**	**Nayan. P. Joshi**
Sem & Sec	**8th Sem A**	**USN:**	**4AL16CS058**

Online Test Summary

Subject	**----------**		
Max. Marks	**----------**	**Score**	**----------**

Certification Course Summary

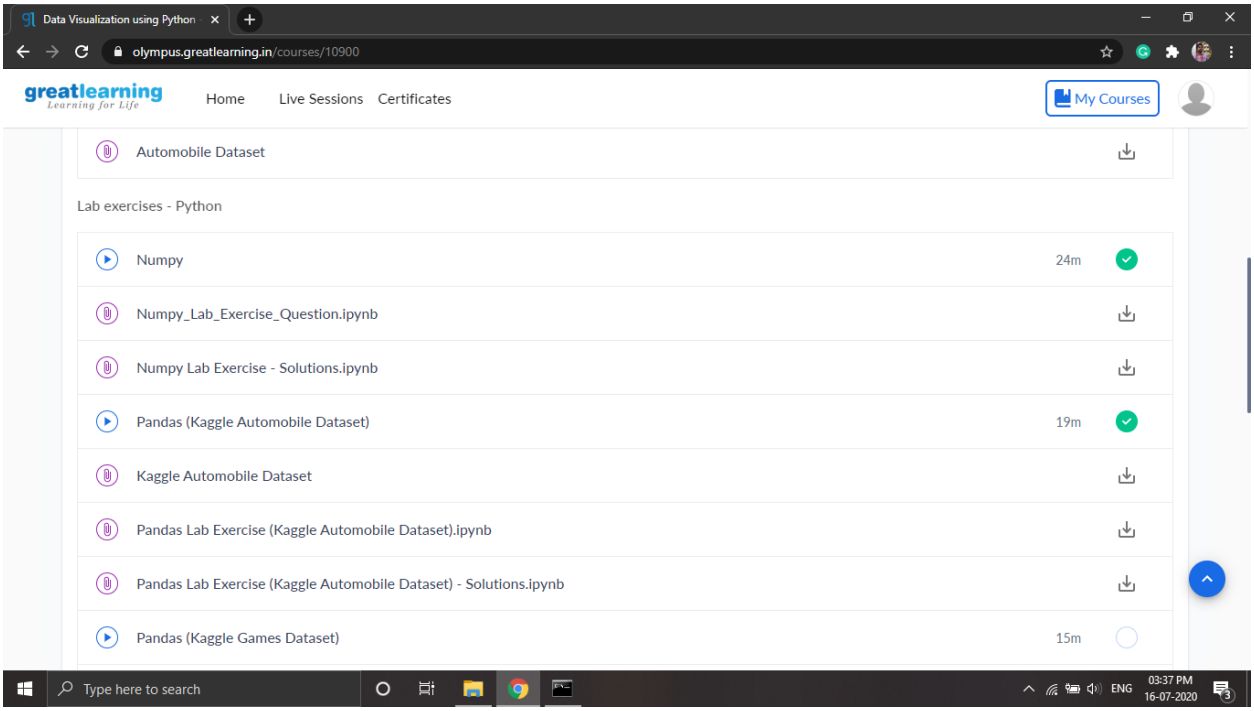
Course	**Data Visualization using python**		
Certificate Provider	**Great learning academy**	**Duration**	**2hrs**

Coding Challenges

Problem Statement: C-Program to check weather a number is Armstrong or Palindrome using user-defined Functions

Status: Solved

Uploaded the report in GitHub	**Yes**
If yes Repository name	**nayan1998**
Uploaded the report in slack	**Yes**


Data Visualization using Python
olympus.greatlearning.in/courses/10900
greatlearning
Learning for Life
Home
Live Sessions
Certificates
My Courses
Automobile Dataset
Lab exercises - Python
Numpy
24m
Numpy_Lab_Exercise_Question.ipynb
Numpy Lab Exercise - Solutions.ipynb
Pandas (Kaggle Automobile Dataset)
19m
Kaggle Automobile Dataset
Pandas Lab Exercise (Kaggle Automobile Dataset).ipynb
Pandas Lab Exercise (Kaggle Automobile Dataset) - Solutions.ipynb
Pandas (Kaggle Games Dataset)
15m
Type here to search
ENG
03:37 PM
16-07-2020

Coding Solution

```c
#include <math.h>
#include <stdio.h>

int checkPrimeNumber(int n);
int checkArmstrongNumber(int n);

int main() {
   int n, flag;
   printf("Enter a positive integer: ");
   scanf("%d", &n);


   flag = checkPrimeNumber(n);
   if (flag == 1)
      printf("%d is a prime number.\n", n);
   else
      printf("%d is not a prime number.\n", n);


   flag = checkArmstrongNumber(n);
   if (flag == 1)
      printf("%d is an Armstrong number.", n);
   else
      printf("%d is not an Armstrong number.", n);
   return 0;
}


int checkPrimeNumber(int n) {
   int i, flag = 1, squareRoot;

   squareRoot = sqrt(n);
   for (i = 2; i <= squareRoot; ++i) {
      if (n % i == 0) {
         flag = 0;
         break;
      }
   }
   return flag;
}
```

```
int checkArmstrongNumber(int num) {
   int originalNum, remainder, n = 0, flag;
   double result = 0.0;


   for (originalNum = num; originalNum != 0; ++n) {
      originalNum /= 10;
   }

   for (originalNum = num; originalNum != 0; originalNum /= 10) {
      remainder = originalNum % 10;


      result += pow(remainder, n);
   }


   if (round(result) == num)
      flag = 1;
   else
      flag = 0;
   return flag;
}
```