DAILY ONLINE ACTIVITIES SUMMARY

Date:	**23/06/2020**	**Name:**	**Deepa**
Sem & Sec	**8th Sem**	**USN:**	**4AL16CS029**
Online Test Summary			
Subject	**- -**		
Max. Marks	**- -**	**Score**	**- -**
Certification Course Summary			
Course	**Complete Python 3 course for Beginners**		
Certificate Provider	**udemy.com/**	**Duration**	**18 hrs**
Coding Challenges			
Problem Statement: 1) Write a C Program to Sort a stack using a temporary stack			
Status: Completed			
Uploaded the report in Github	**Yes**		
If yes Repository name	**Daily_report**		
Uploaded the report in slack	**yes**		

Online Test Details:

- -

Certification Course Details:

- [x] 125. Qt key shortcuts
  10min
- [x] 126. QtSplitter and section wrap up
  10min

Section 13: Project #9 - Data Visualization
9 / 12 | 1hr 15min

- [x] 127. Installing matplotlib
  3min
- [x] 128. World population graph
  7min
- [x] 129. Adding labels and custom line color
  6min
- [x] 130. Multiple lines and line styling
  4min
- [x] 131. Configuring the graph
  4min
- [x] 132. Let's make pie (charts)
  6min
- [x] 133. Letting Pandas make data simpler
  9min
- [x] 134. Using Panda's data for pie charts
  4min
- [x] 135. Bar charts pt 1

Coding Challenges Details:

Program 1:

```
#include<stdio.h>
int stack[100], temps[100], temp, choice, n, top, ttop, x, i;
void push(int x)
{
   top++;
   stack[top]=x;
}
void pop()
{
   temp = stack[top];
   top--;
}
void display()
{
   if(ttop>=0)
   {
      printf("\n The sorted elements in STACK \n");
      for(i=ttop; i>=0; i--)
         printf("\n%d",temps[i]);
   }
```

```
    else
    {
        printf("\n The STACK is empty");
    }

}
int main()
{
    top=-1;
    ttop = -1;
    printf("\n Enter the size of STACK[MAX=100]:");
    scanf("%d",&n);
    printf("Enter the elements in the stack:\n");
    for(i=0; i<n; i++)
    {
        printf(" Enter a value to be pushed:");
        scanf("%d",&x);
        push(x);
    }

    while(top != -1)
    {
        pop();
```

```
        while(ttop != -1 && temps[ttop] > temp)
        {
            push(temps[ttop]);
            ttop--;


        }
        ttop++;
        temps[ttop] = temp;
    }
    display();
}
```