DAILY ONLINE ACTIVITIES SUMMARY

Date:	**19/05/2020**	**Name:**	**Prathiksha**
Sem & Sec	**8th sem & B sec**	**USN:**	**4AL16CS070**

Online Test Summary

Subject	**Big Data Analytics(BDA)**		
Max. Marks	**30**	**Score**	**9**

Certification Course Summary

Course	**Introduction To Ethical Hacking**		
Certificate Provider	**Great Learning Academy**	**Duration**	**6hrs**

Coding Challenges

Problem Statement:

1. Find out what will be shortest palindrome string.
2. Write a simple code to identify given linked list is palindrome or not by using stack.

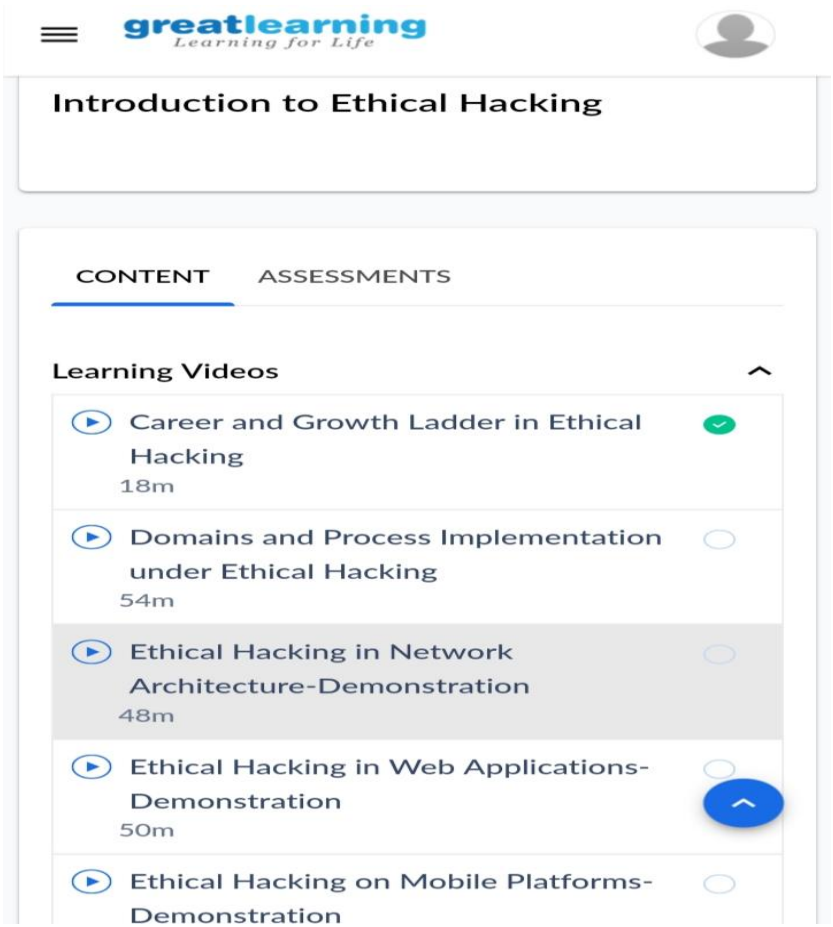
Status: Solved

Uploaded the report in Github	**Yes**
If yes Repository name	**Online coding challenges**
Uploaded the report in slack	**Yes**

Online Test Details:

IA1 portion was Module 1.

Certification Course Details:



Topic : Introduction to Ethical Hacking

Coding Challenges Details:

Program 1:

```
import java.util.Scanner;
        public class ShortestPalindromeDemo {
        public static String shortestPalindrome(String str) {
        int x=0;
        int y=str.length()-1;
        while(y>=0){
        if(str.charAt(x)==str.charAt(y)){
        x++;
        }
        y--;
        }
        if(x==str.length())
        return str;
        String suffix = str.substring(x);
        String prefix = new StringBuilder(suffix).reverse().toString();
        String mid = shortestPalindrome(str.substring(0, x));
        return prefix+mid+suffix;
        }
        public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.println("Enter a String to find out shortest palindrome");
        String str=in.nextLine();
        System.out.println("Shortest palindrome of "+str+" is "+shortestPalindrome(str));
        }
```

Program 2:

```
        import java.util.Stack;
        class Node {
        int data;
        Node next;
        Node(int i)
        {
        this.data = i;
        this.next = null;
        }
        };
        class Main
        {
        public static boolean isPalindrome(Node head)
        {
        Stack s = new Stack<>();
        Node node = head; while (node != null) {      s.push(node.data);       node = node.next; }
        node = head; while (node != null) {    int top = s.pop(); if (top != node.data) {
                  return false;     } node = node.next; }
        }
        public static void main(String[] args)
        {
        Node head = new Node(1);
```

```
head.next = new Node(2);
head.next.next = new Node(3);
head.next.next.next = new Node(2);
head.next.next.next.next = new Node(1);
if (isPalindrome(head)) {        System.out.print("Linked List is a palindrome."); } else {
          System.out.print("Linked List is not a palindrome."); }
}
}
```