DAILY ONLINE ACTIVITIES SUMMARY

Date:	20/5/2020	**Name:**	Vleena Mascarenhas
Sem & Sec	8th & B	**USN:**	4AL16CS121

Online Test Summary

Subject	Internet of Things		
Max. Marks	30	**Score**	22

Certification Course Summary

Course	Introduction to Ethical Hacking		
Certificate Provider	Great learning academy	**Duration**	6hrs

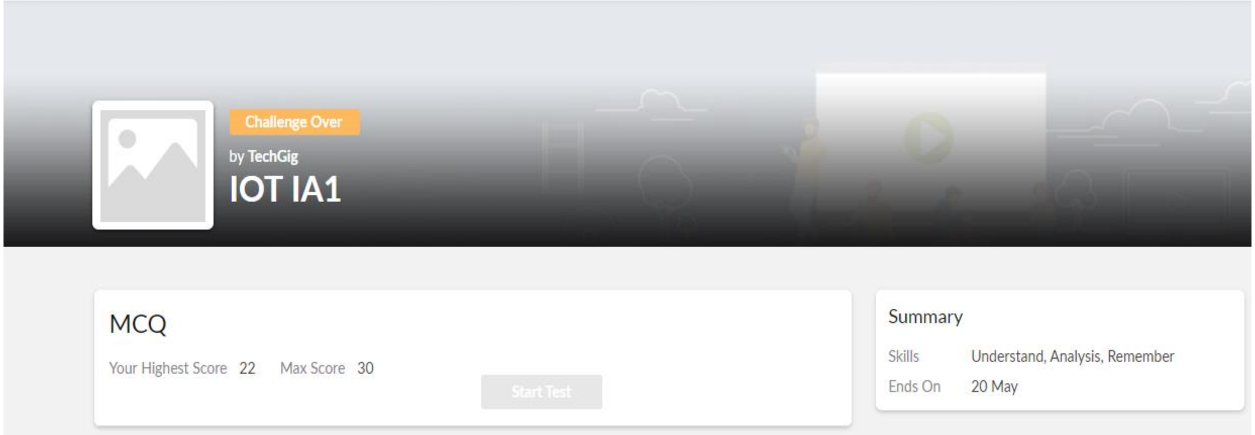
Coding Challenges

Problem Statement:
1. Write a C Program to Reverse a Linked List in groups of given size.

Status: Solved

Uploaded the report in Github	yes
If yes Repository name	vleena
Uploaded the report in slack	yes

Online Test Details: (Attach the snapshot and briefly write the report for the same)



by TechGig

IOT IA1

MCQ

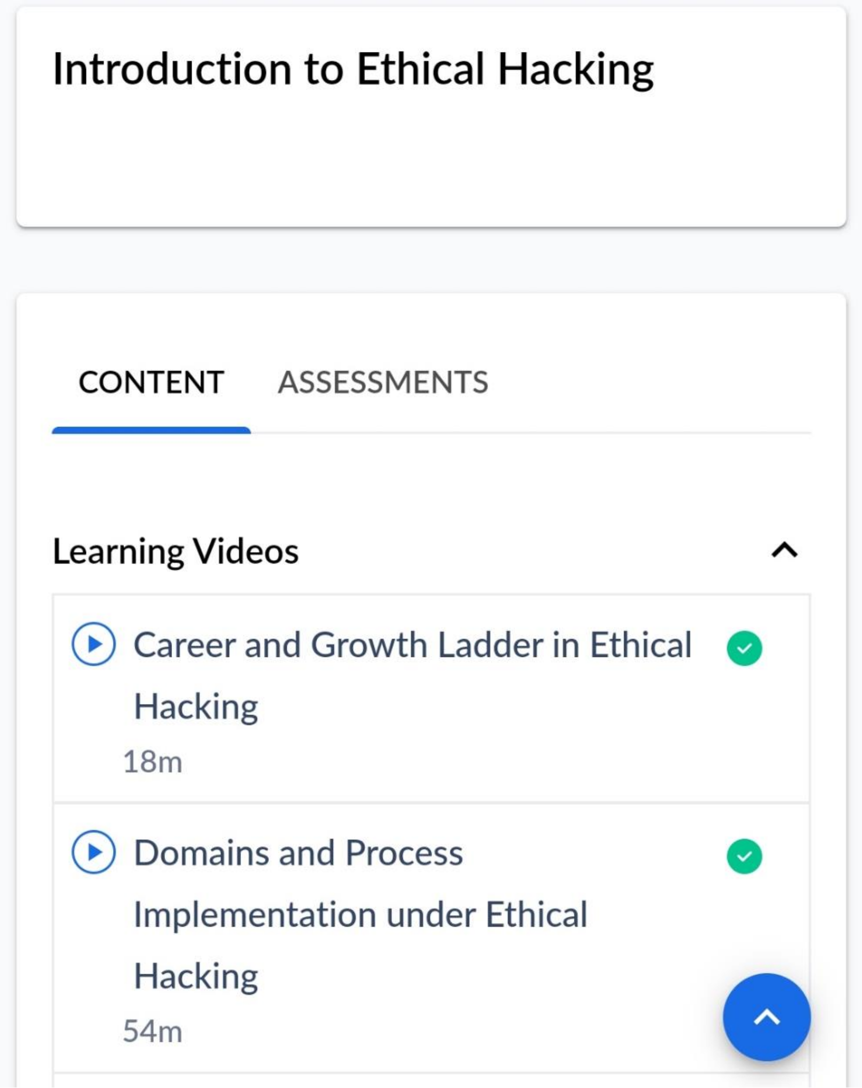
Your Highest Score 22 Max Score 30

Summary

Skills Understand, Analysis, Remember

Ends On 20 May

Certification Course Details: (Attach the snapshot and briefly write the report for the same)



Introduction to Ethical Hacking

Learning Videos

Career and Growth Ladder in Ethical Hacking ✓

18m

Domains and Process Implementation under Ethical Hacking ✓

54m

Coding Challenges Details: (Attach the snapshot and briefly write the report for the same)

```
#include<stdio.h>
#include<stdlib.h>
struct Node
{
int data;
struct Node* next;
};
struct Node reverse (struct Node head, int k)
{
struct Node current = head;
struct Node next = NULL;
struct Node prev = NULL;
int count = 0;

while (current != NULL && count < k)
{
next = current->next;
current->next = prev;
prev = current;
current = next;
 count++;
 }
if (next != NULL) {
head->next = reverse(next, k);
return prev;
```

```
}

void push(struct Node** head_ref, int new_data)
{
struct Node* new_node =(struct Node*) malloc(sizeof(struct Node));

new_node->data = new_data;
new_node->next = (*head_ref);
(*head_ref) = new_node;

}

void printList(struct Node *node)
{
while (node != NULL)
{
printf("%d ", node->data);
node = node->next;
}
}

int main(void)
{
struct Node* head = NULL;
push(&head, 8);
push(&head, 7);
push(&head, 6);

push(&head, 5);

push(&head, 4);

push(&head, 3);

push(&head, 2);

push(&head, 1);


printf("\nGiven linked list \n");

printList(head);

head = reverse(head, 2);

printf("\nReversed Linked list \n");

printList(head);

 return(0);


}
```