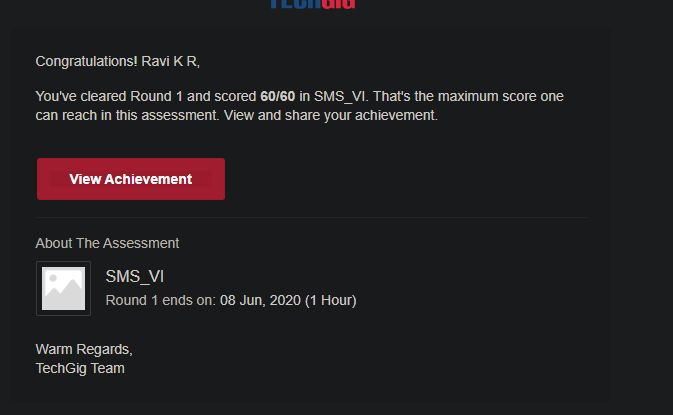
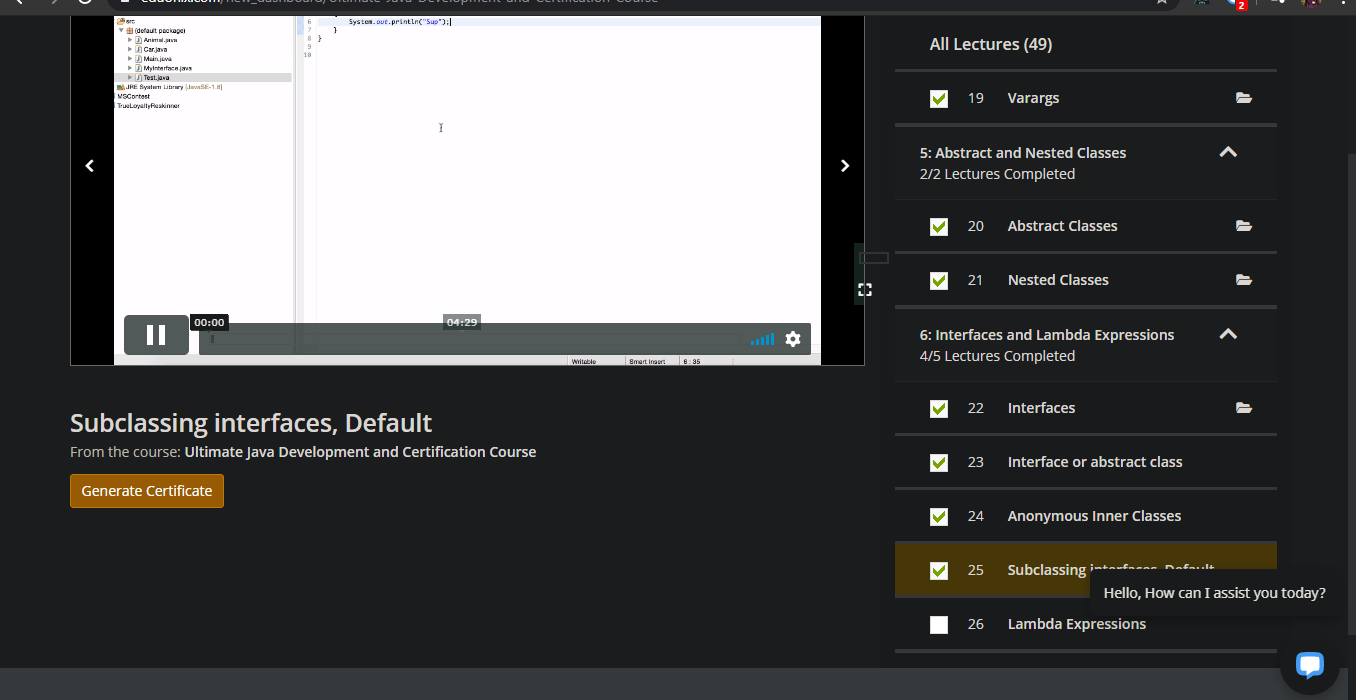
**DAILY ONLINE ACTIVITIES SUMMARY**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Date:** | **8/06/2020** | | | | **Name:** | **Ravi K R** | |
| **Sem & Sec** | **8th- B** | | | | **USN:** | **4AL16CS076** | |
| **Online Test Summary** | | | | | | | |
| **Subject** | | **SMS** | | | | | |
| **Max. Marks** | | **60** | | **Score** | | **60** | |
| **Certification Course Summary** | | | | | | | |
| **Course** | **Ultimate Java Development and Certification Course** | | | | | | |
| **platform** | | | **edunix** | **Duration** | | | **10 hrs** |
| **Coding Challenges** | | | | | | | |
| **Problem Statement:**  **You are given two non-empty linked lists representing two non-negative integers. The digits are stored in reverse order and each of their nodes contain a single digit. Add the two numbers and return it as a linked list.**  **You may assume the two numbers do not contain any leading zero, except the number 0 itself.**  **Example:**  **Input: (2 -> 4 -> 3) + (5 -> 6 -> 4)**  **Output: 7 -> 0 -> 8**  **Explanation: 342 + 465 = 807.**  **and return it as a linked list.** | | | | | | | |
| **Status: Executed** | | | | | | | |
| **Uploaded the report in Github** | | | | **Yes** | | | |
| **If yes Repository name** | | | | <https://github.com/alvas-education-foundation/Ravi_kr> | | | |
| **Uploaded the report in slack** | | | | **Yes** | | | |

Online Test Details:



Certification:



Coding Challenges Details:

import java.util.\*;

/\*\*

\* Definition for singly-linked list.

\* public class ListNode {

\* int val;

\* ListNode next;

\* ListNode() {}

\* ListNode(int val) { this.val = val; }

\* ListNode(int val, ListNode next) { this.val = val; this.next = next; }

\* }

\*/

class Solution {

Solution Linklist=new Solution();

ListNode head;

String list1="",list2="";

public ListNode addTwoNumbers(ListNode l1, ListNode l2) {

ListNode ptr=l1;

while(ptr!=null){

list1=list1+Integer.toString(ptr.val);

ptr=ptr.next;

}

ListNode ptr2=l2;

while(ptr2!=null){

list2=list2+Integer.toString(ptr2.val);

ptr2=ptr2.next;

}

char ch[]=list1.toCharArray();

String rev1="";

for(int i=ch.length-1;i>=0;i--){

rev1+=ch[i];

}

char ch2[]=list2.toCharArray();

String rev2="";

for(int i=ch2.length-1;i>=0;i--){

rev2+=ch2[i];

}

int a=Integer.parseInt(rev1);

int b=Integer.parseInt(rev2);

int sum=a+b;

String sum1=Integer.toString(sum);

char ch3[]=sum1.toCharArray();

String rev3="";

for(int i=ch3.length-1;i>=0;i--){

rev3+=ch3[i];

}

char rev4[]=rev3.toCharArray();

for(int i=0;i<rev4.length;i++){

int m=Integer.parseInt(String.valueOf(rev4[i]));

Linklist.head= new ListNode(m);

head=head.next;

System.out.println(head.val);

}

// System.out.println(head);

return head;

}

}