DAILY ONLINE ACTIVITIES SUMMARY

Date:	15/6/2020	**Name:**	Vleena Mascarenhas
Sem & Sec	8th & B	**USN:**	4AL16CS121

Online Test Summary

Subject	System Modeling and Simulation		
Max. Marks	60	**Score**	Mail Not Received

Certification Course Summary

Course	Introduction to Information Security		
Certificate Provider	Great learning academy	**Duration**	5.5hrs

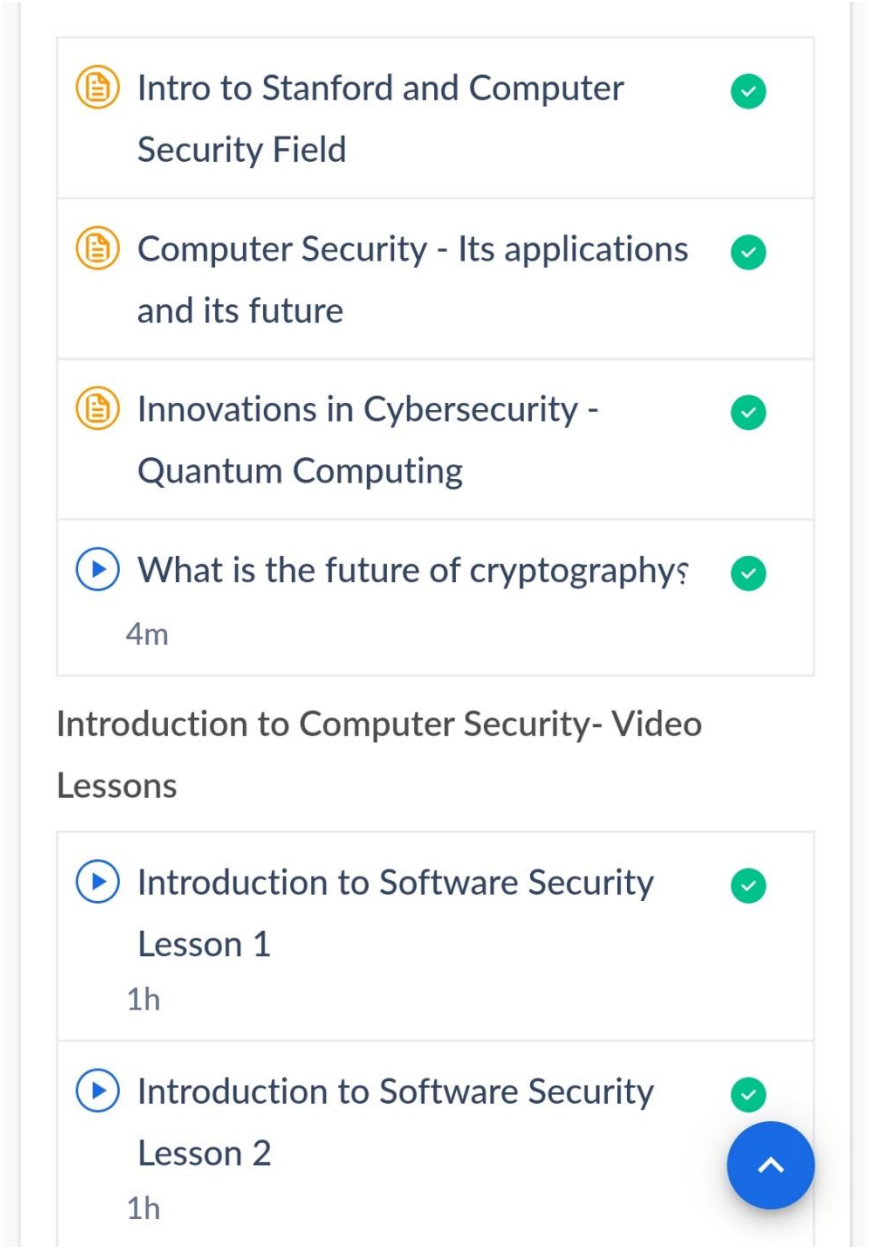
Coding Challenges

Problem Statement:
Java program to implement triply linked list.

Status: Solved

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

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



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

Problem Statement:

Java program to implement triply linked list.

```
import java.util.Scanner;
class TLLNode
{
	TLLNode left,right,middle;
	int data;
	public TLLNode(int x)
	{
		data=x;
		left=null;
		right=null;
		middle=null;
	}
}

class TriplyLinkedList
{
	TLLNode root,tmp;
	public TriplyLinkedList()
	{
		root=null;
		tmp=null;
```

```
}
public Boolean isEmpty()
{
        return root==null;
}
public void makeEmpty()
{
        root=null;
        tmp=null;
}
public void insert(int x)
{
        root=insert(root,x);
}
public TLLNode insert(TLLNode r,int x)
{
        if(r==null)
        {
                r=new TLLNode(x);
                r.middle=tmp;
        }
        else
        {
                tmp=r;
                if(r.data>=x)
```

```
                r.left=insert(r.left,x);
                else
                r.right=insert(r.right,x);
        }
return r;
}
public void printList()
{
        printList(root);
}
private void printList(TLLNode r)
{
        if(r!=null)
        {
                printList(r.left);
                System.out.print(r.data+" ");
                printList(r.right);
                }
        }
}
public class TriplyLinkedListTest
{
        public static void main(String[] args)
        {
                Scanner scan=new Scanner(System.in);
```

```
System.out.println(“Triply Linked List Test\n”);
TriplyLinkedListTest tll=new TriplyLinkedListTest();
char ch;
do
{
        System.out.println(“\nTriply Linked List Operations\n”);
        System.out.println(“1.insert”);
        System.out.println(“2.check empty”);
        System.out.println(“3.make empty”);
        int choice=scan.nextInt();
        switch(choice)
        {
        case 1:
                System.out.println(“Enter integer element to insert”);
                tll.insert(scan.nextInt());
                break;
        case 2:
                System.out.println(“Empty status = “+ tll.isEmpty());
                break;
        case 3:
                System.out.println(“\nList Cleared\n”);
                tll.makeEmpty();
                break;
        default:
                System.out.println(“Wrong Entry\n”);
```

```
                    break;
                }
                System.out.println("\nList:");
                tll.printList();
                System.out.println("\nDo you want to continue (type y or n) \n");
                ch=scan.next().charAt(0);
                }while (ch == 'Y' || ch == 'y');
            }
}
```

OUTPUT:

Triply Linked List Test

Triply Linked List Operations

1.insert

2.check empty

3.make empty

1

Enter integer element to insert

97

List: 97

Do you want to continue (type y or n)

y

Triply Linked List Operations

1.insert

2.check empty

3.make empty

1

Enter integer element to insert

24

List:24 97

Do you want to continue (type y or n)

y

Triply Linked List Operations

1.insert

2.check empty

3.make empty

1

Enter integer element to insert

6

List: 6 24 97

Do you want to continue (type y or n)

y

Triply Linked List Operations

1.insert

2.check empty

3.make empty

1

Enter integer element to insert

19

List: 6 19 24 97

Do you want to continue (type y or n)

y

Triply Linked List Operations

1.insert

2.check empty

3.make empty

1

Enter integer element to insert

94

List: 6 19 24 94 97

Do you want to continue (type y or n)

y

Triply Linked List Operations

1.insert

2.check empty

3.make empty

1

Enter integer element to insert

57

List: 6 19 24 57 94 97

Do you want to continue (type y or n)

y

Triply Linked List Operations

1.insert

2.check empty

3.make empty

1

Enter integer element to insert

23

List: 6 19 23 24 57 94 97

Do you want to continue (type y or n)

y

Triply Linked List Operations

1.insert

2.check empty

3.make empty

2

Empty status = false

List: 6 19 23 24 57 94 97

Do you want to continue (type y or n)

y

Triply Linked List Operations

1.insert

2.check empty

3.make empty

3

List Cleared

List:

Do you want to continue (type y or n)

y

Triply Linked List Operations

1.insert

2.check empty

3.make empty

2

Empty status = true

List:

Do you want to continue (type y or n)

n