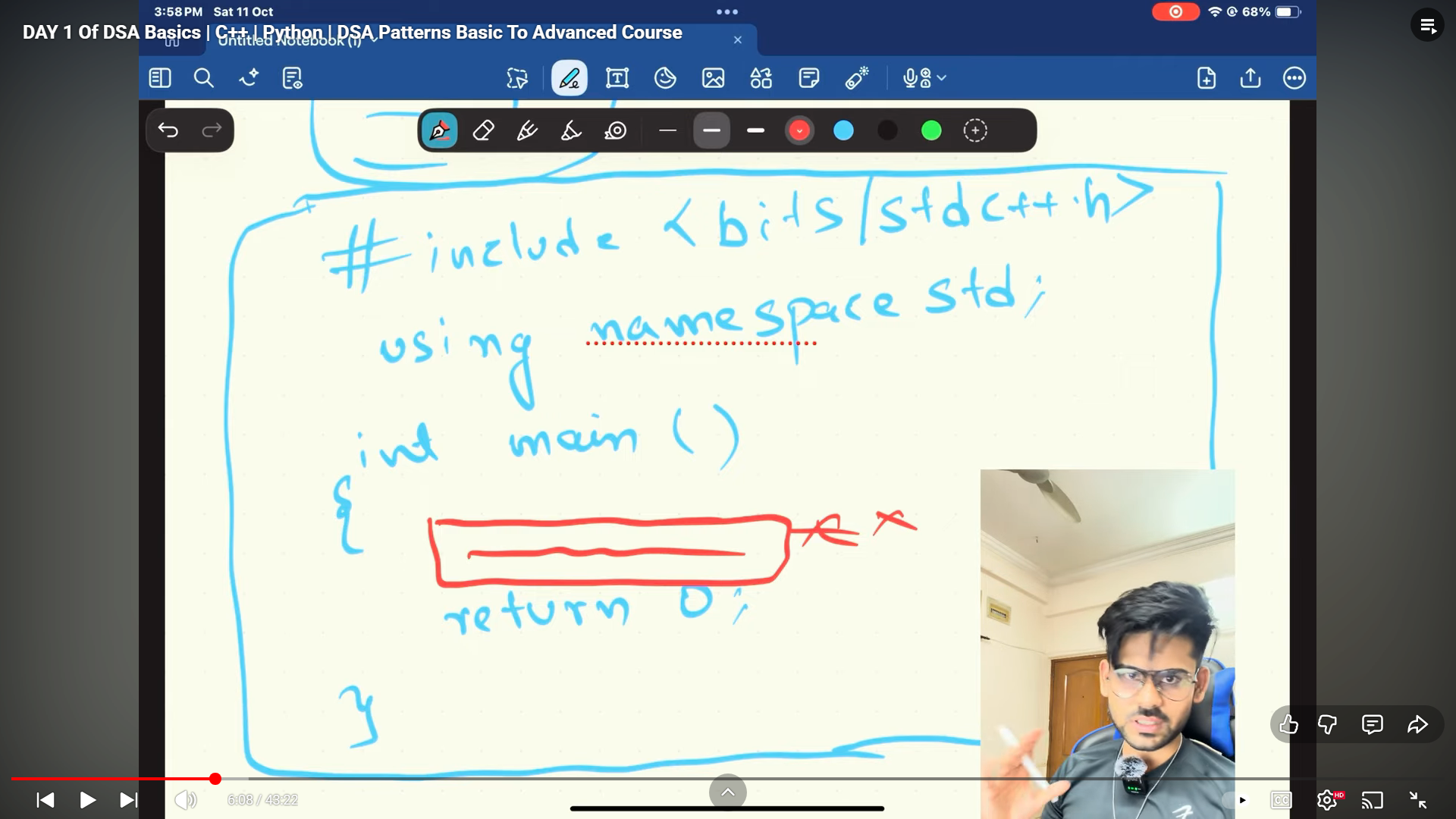
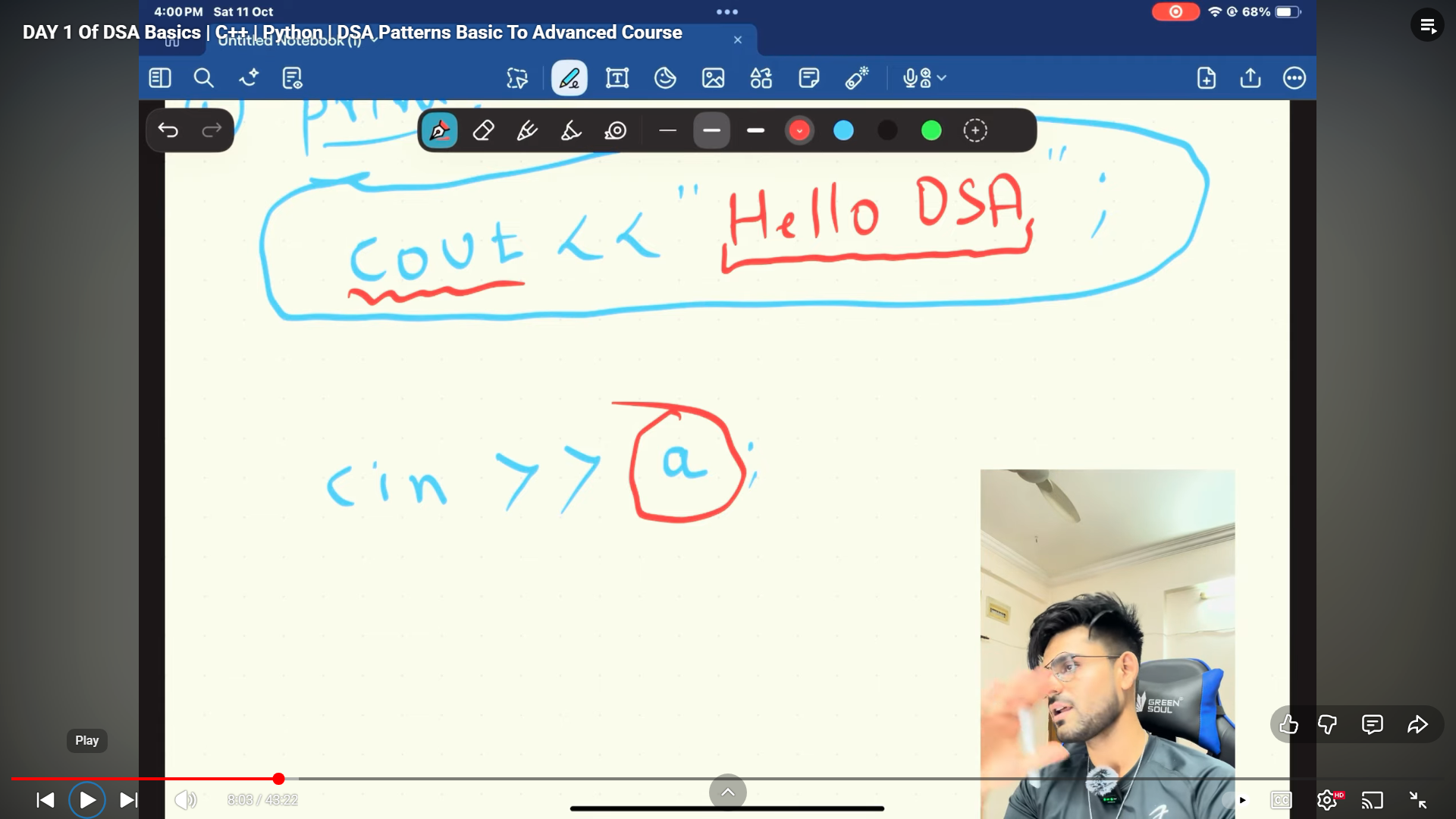
**Basic Understanding of DSA before beginning with Patterns –**





**Bssic template –**

#include<bits/stdc++.h>

using namespace std;

int main()

{

cout<<"I will crack 20 LPA Package soon"<<endl;

return 0;

}

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

int main()

{

    int a, b, c;

    cout << "Enter respectively all three values - " << endl;

    cin >> a >> b >> c;

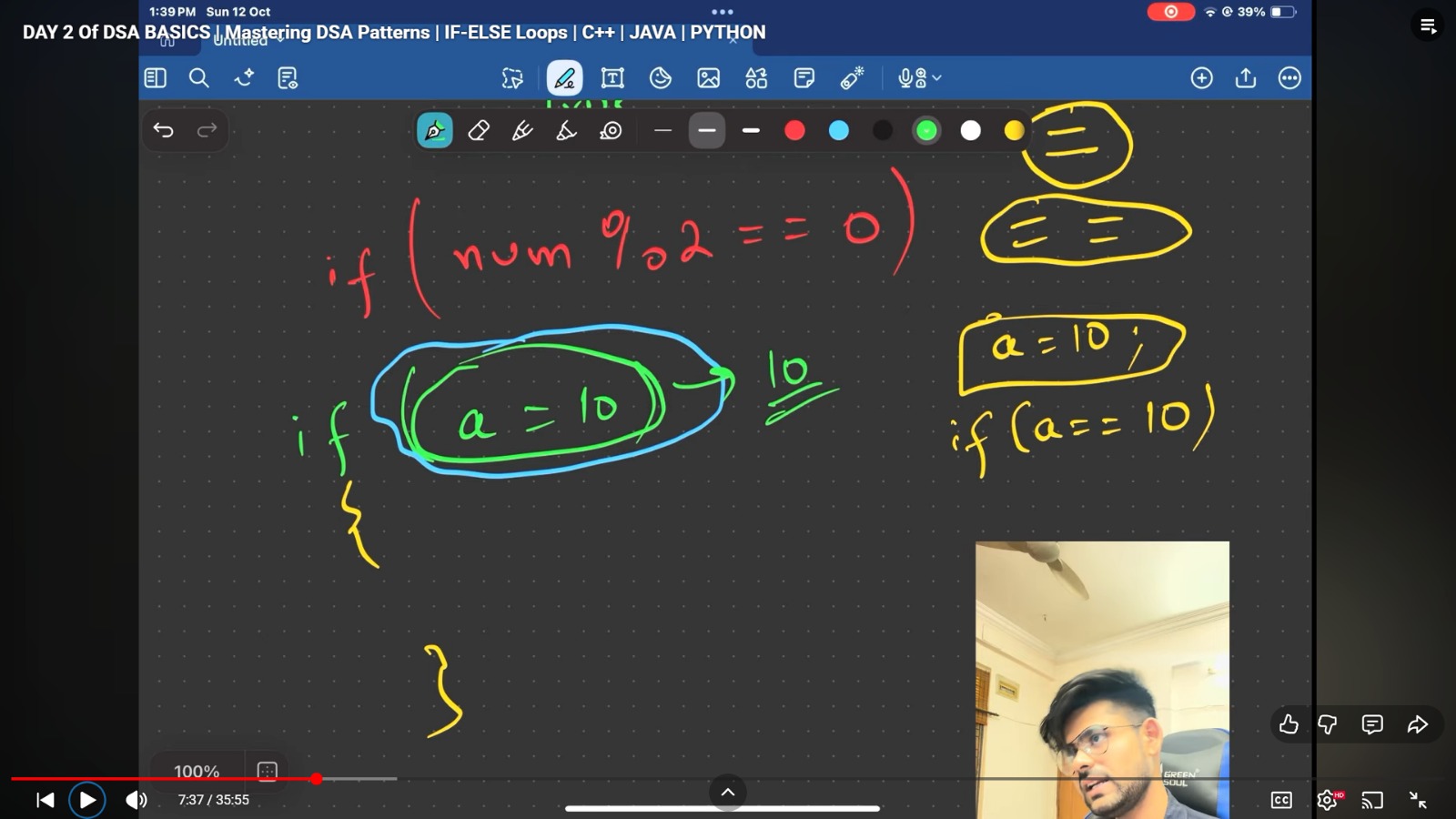
    cout << "So, values are - " << a << "," << b << "," << c << endl;

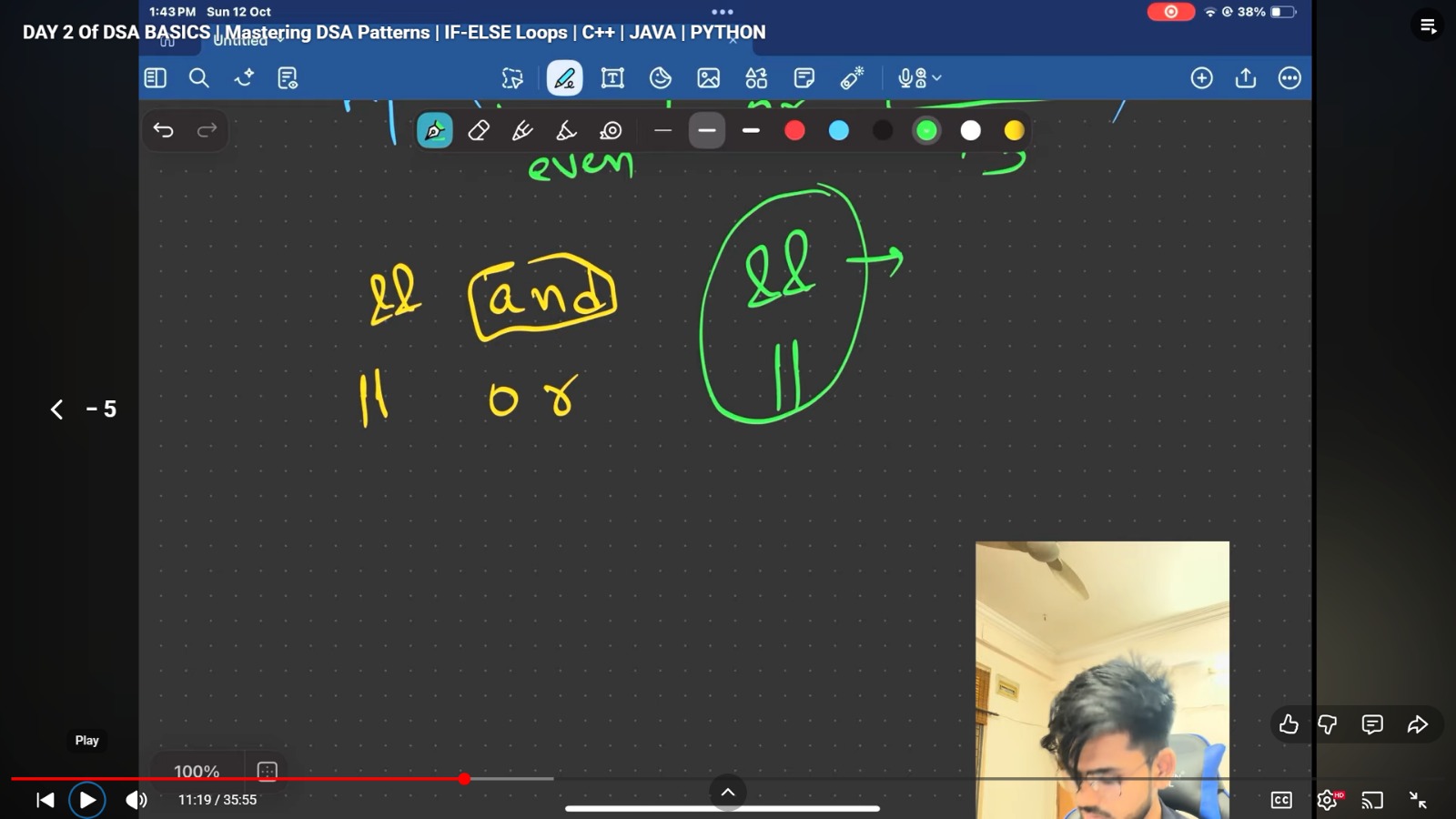
    int product = a \* b \* c;

    cout << "The Product is - " << product << endl;

}

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_





**If Else Understanding -**

int main()

{

    int n = 10;

    if (n % 2 == 0)

    {

        cout << "It's Even" << endl;

    }

    else

    {

        cout << "Odd Number" << endl;

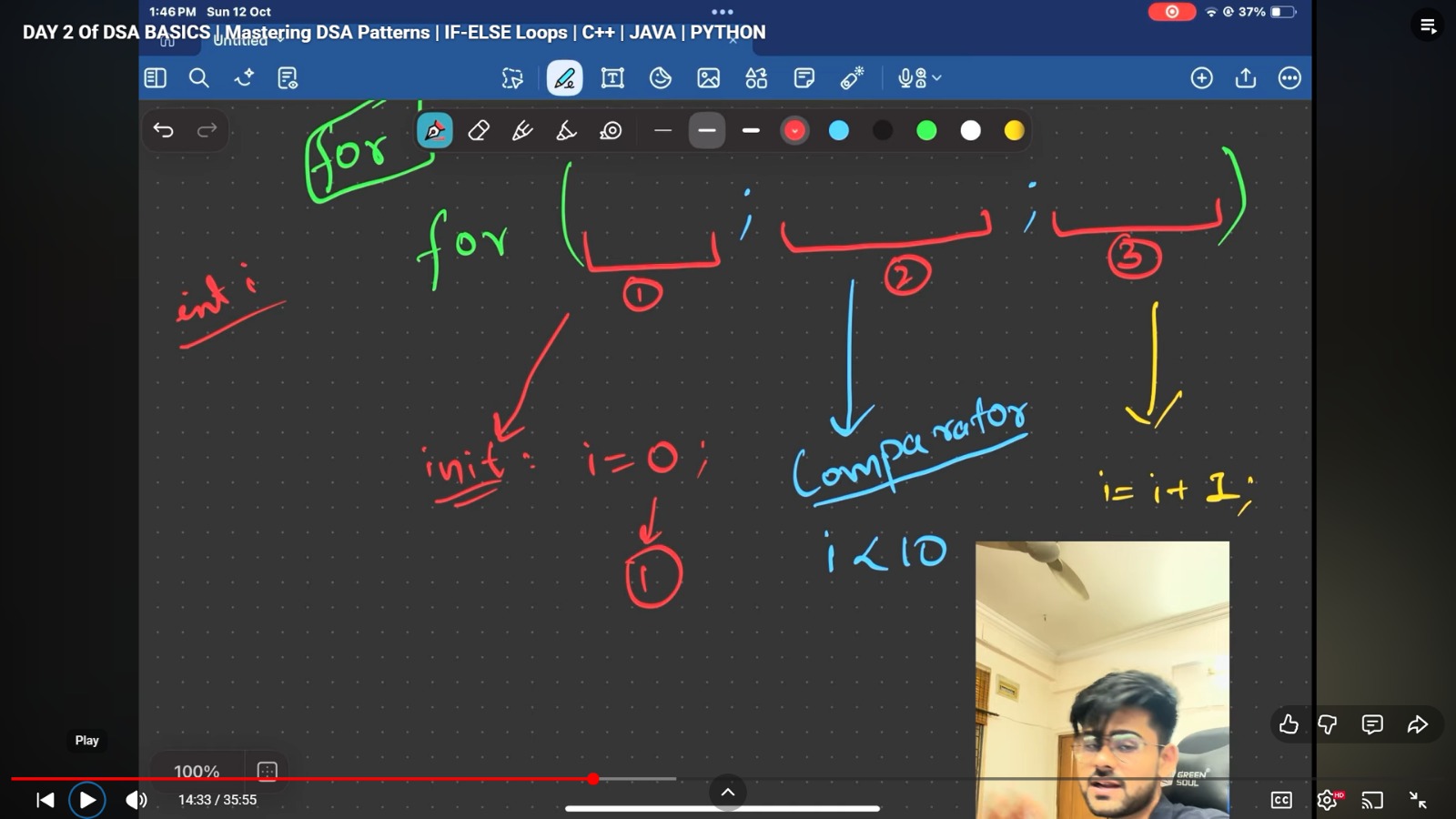
    }

    return 0;

}

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**For loop working –**



int main()

{

    for (int i = 0; i < 5; i++)

    {

        cout << "I am working in Wells Fargo and earning 20 LPA" << endl;

    }

    return 0;

}

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

In CPP – we can write – AND,OR yaa phir symbols also - &&, || but while in java we can only go for symbols - &&,||.

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Qun – Sum of first 10 numbers**

int main()

{

    int sum = 0;

    for (int i = 1; i <= 10; i++)

    {

        sum += i;

    }

    cout << "Sum is - " << sum << endl;

    return 0;

}

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Qun – Sum of first 10 Even number**

int main()

{

    int sum = 0;

    for (int i = 1; i <= 20; i++)

    {

        if (i % 2 == 0)

        {

            sum += i;

        }

    }

    cout << "Sum is - " << sum << endl;

    return 0;

}

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Qun – Sum of Even number b/w 20-40**

int main()

{

    int sum = 0;

    for (int i = 20; i <= 40; i++)

    {

        if (i % 2 == 0)

        {

            sum += i;

        }

    }

    cout << "Sum is - " << sum << endl;

    return 0;

}

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Functions –**

int shubhamMahajan(int n)

{

    int doublework = 2 \* n;

    return doublework;

}

int main()

{

    int m = 10;

    cout << shubhamMahajan(m) << endl;

}

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

class Solution {

public:

int findSum(int n) {

if(n==0)

{

return 0;

}

int sum = (n\*(n+1)/2);

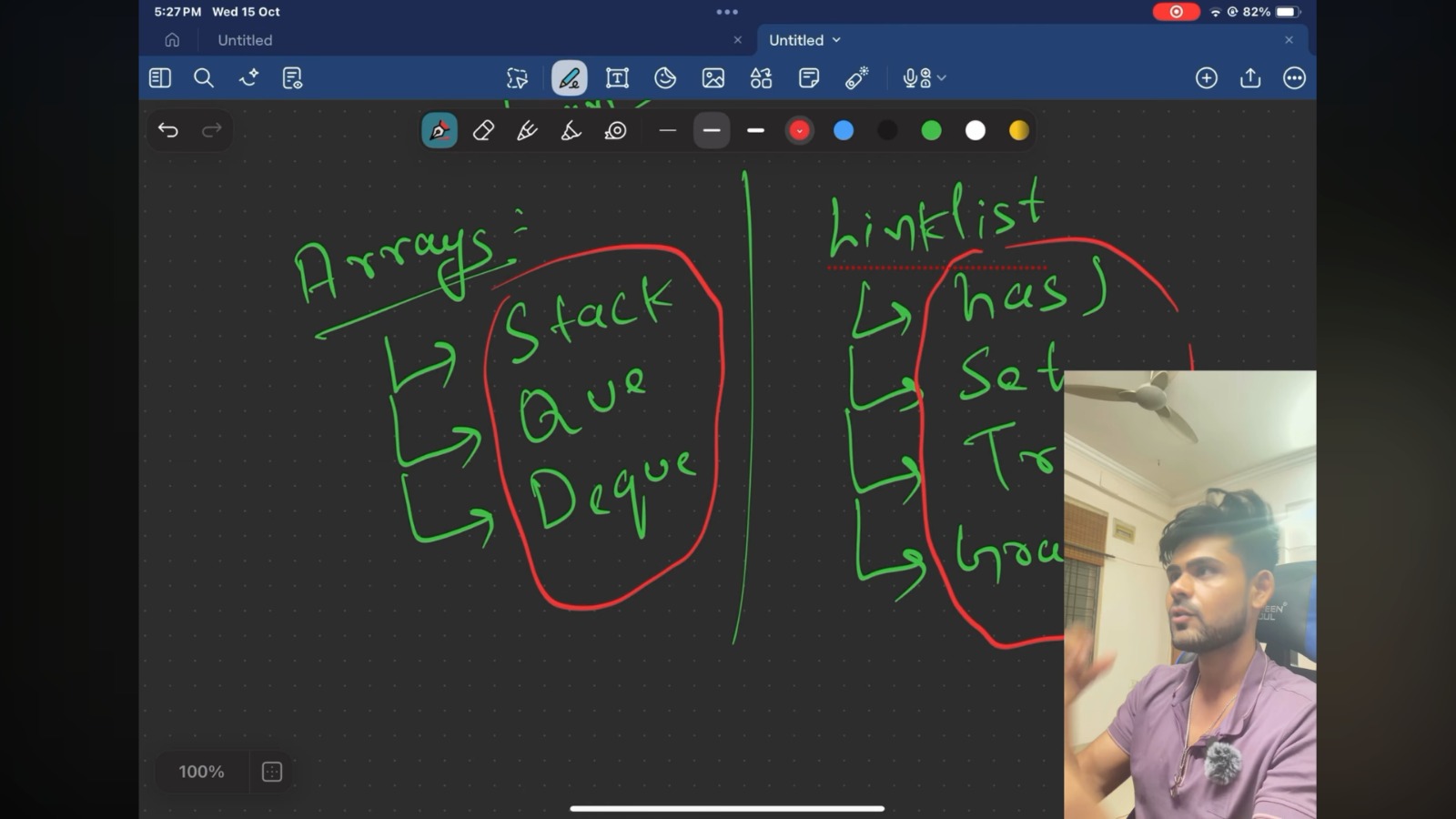
return sum;

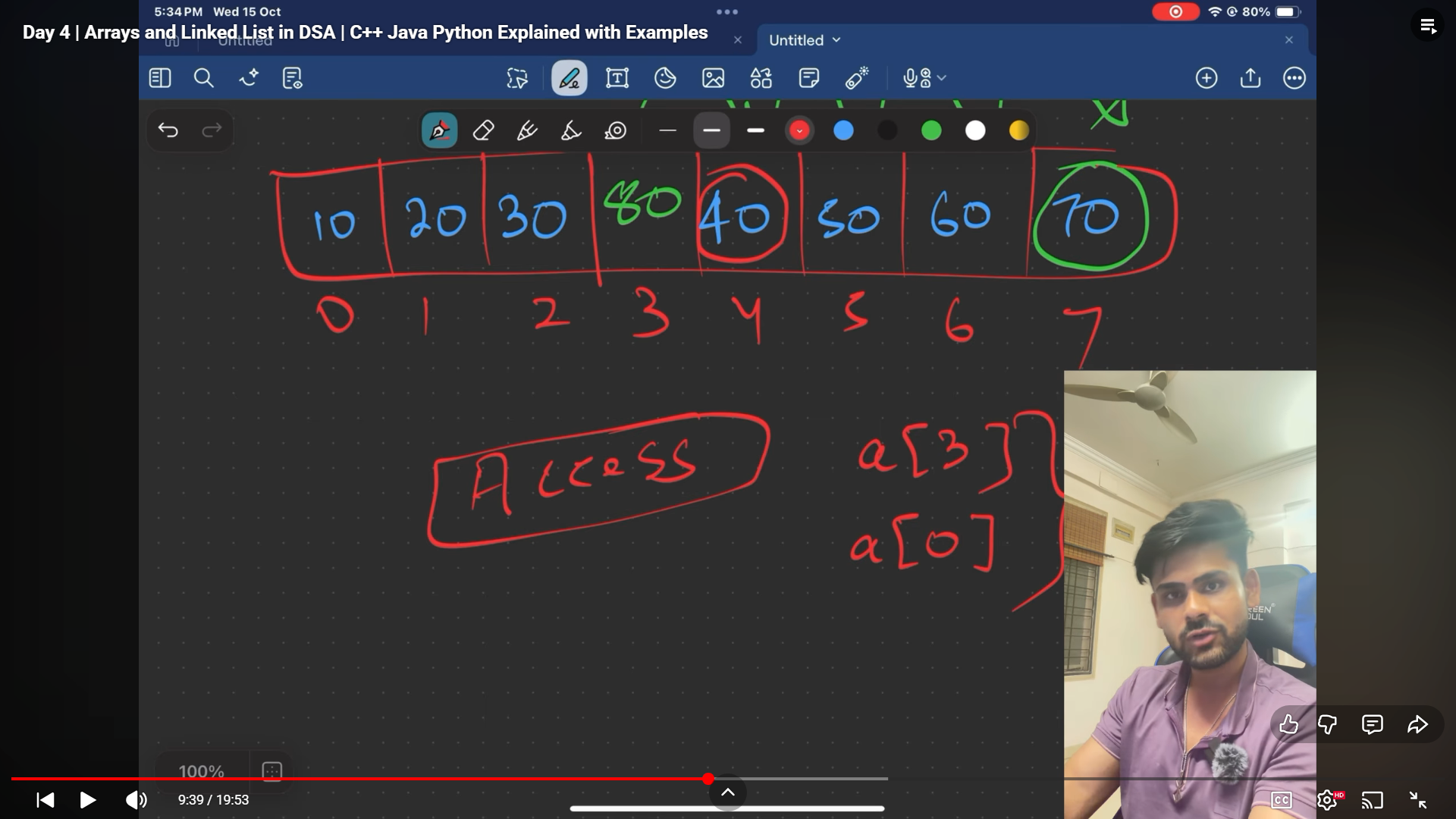
}

};

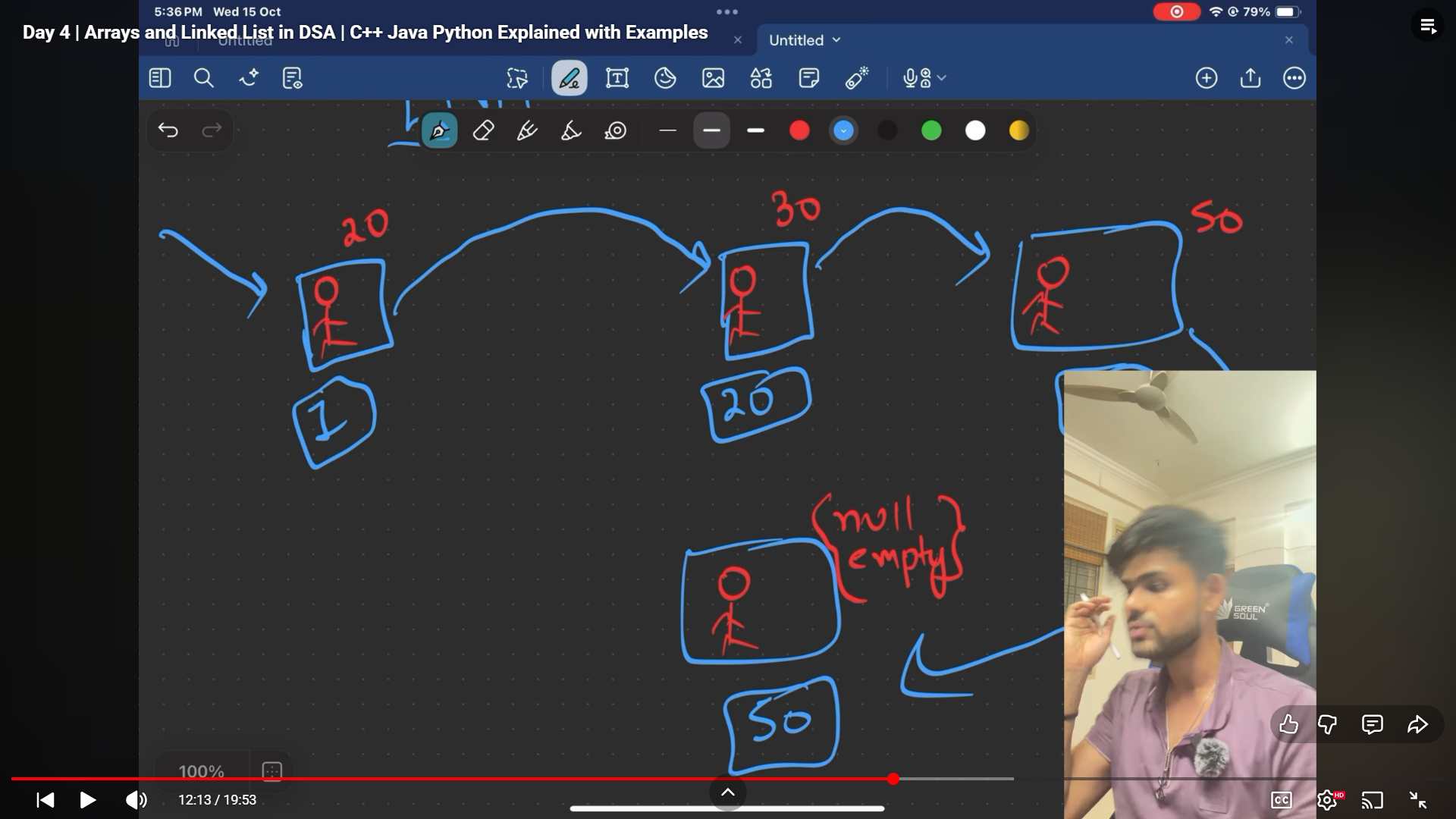
//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Array & Linked List basic Understanding –**





Array can be only worked when data is continues, what if the ewlements are available randomly and need to sit them together –

Can be done by linkedlist, if the address of atleast one nextone is known 

Tradeoffs b/w array & LL -

1. Random Access – always easily accessible because all are continuioes, but in LL, can’t determine because nextone’s address need to inform or only known by previousone. For go to 59 on LL need to go till 58 but in array can be directly accessed.
2. While in array – insertion,deletion bich me is very difficulott… but it’s very easy in LL



