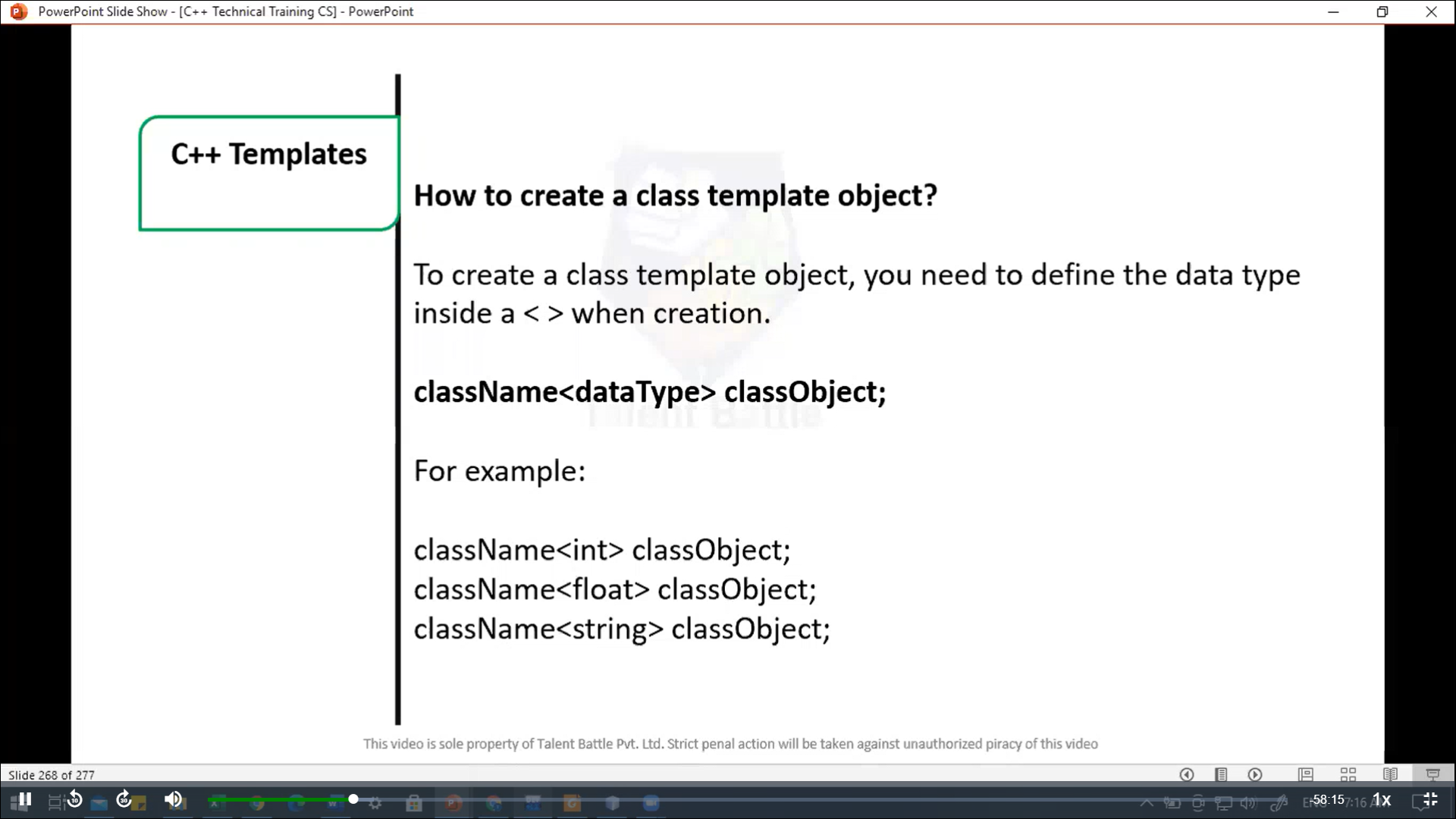
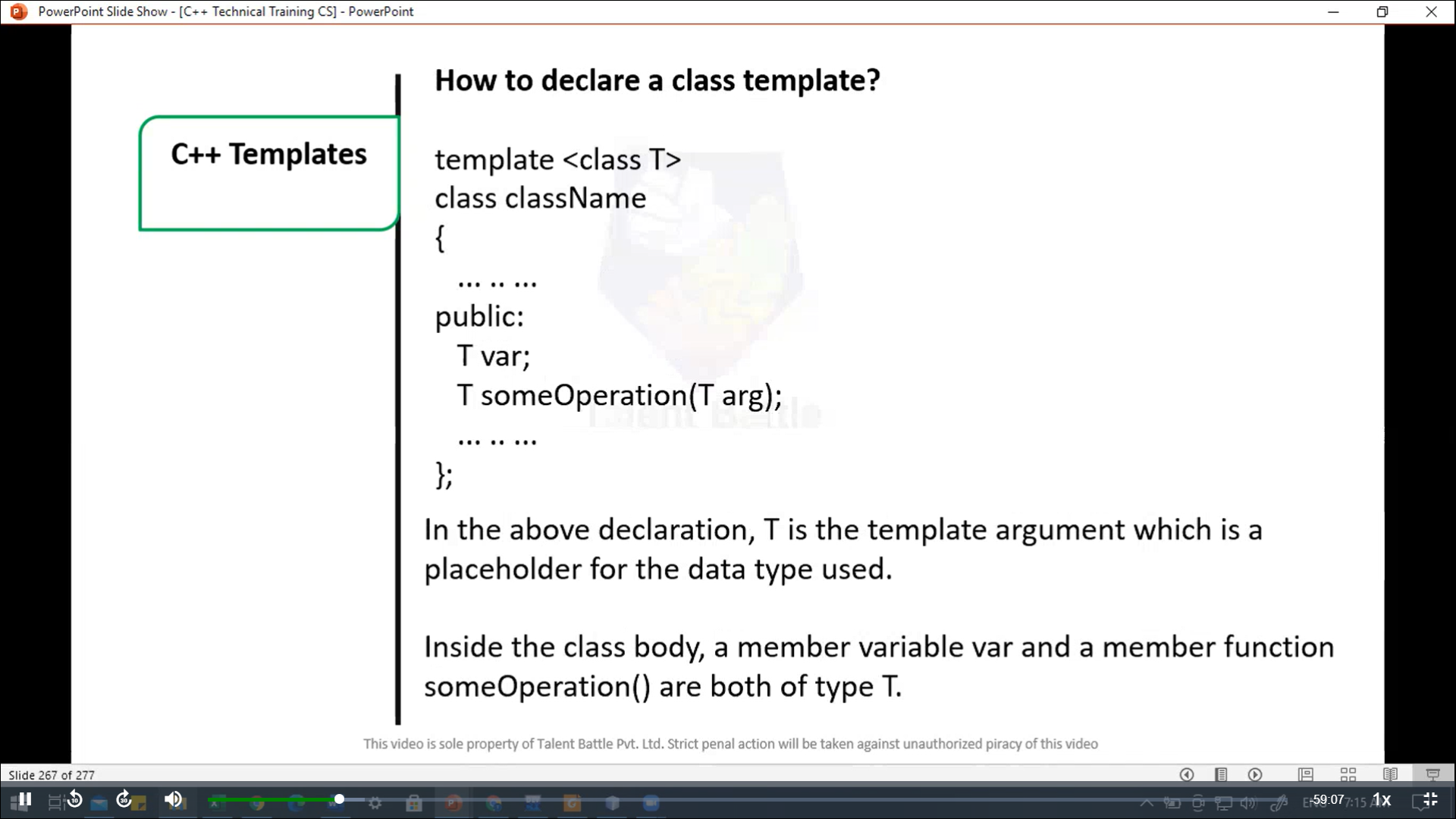
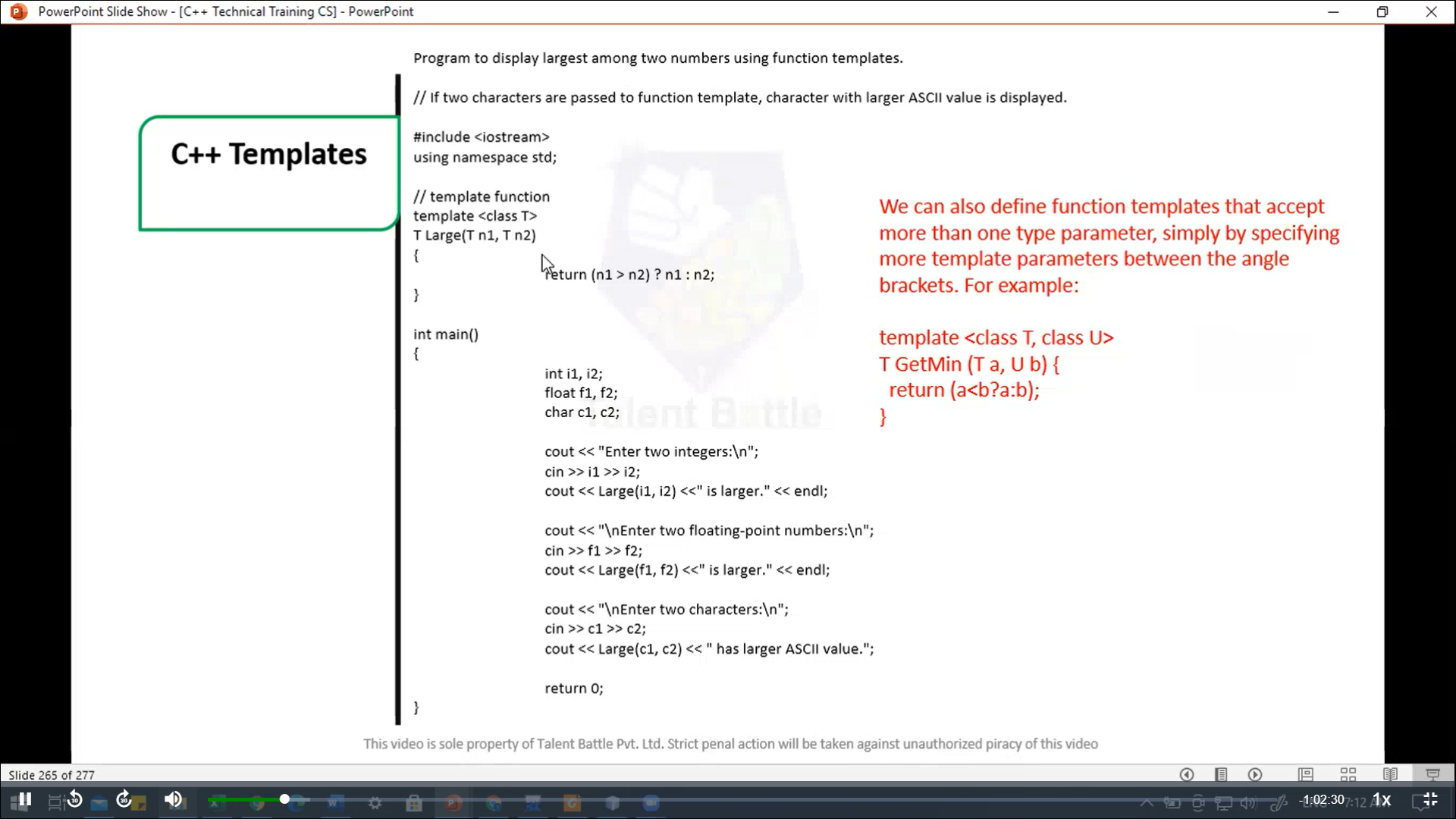
**Day 9**

C++ STL



// Template

// Program to add, subtract, multiply and  divide two numbers using class template

#include<iostream>

using namespace ***std***;

template<class T>

*class* Calculator{

  private***:***

    T num1***,*** num2;

  public***:***

*Calculator*(T ***n1,*** T ***n2***){

      num1 = n1;

      num2 = n2;

    }

    void *displayResult*(){

      cout ***<<*** "Number are: " ***<<*** num1 ***<<*** " and " ***<<*** num2 ***<<*** "." ***<<*** endl;

      cout ***<<*** "Addition is: " ***<<*** *add*() ***<<*** endl;

      cout ***<<*** "Subtraction is: " ***<<*** *subtract*() ***<<*** endl;

      cout ***<<*** "Product is: " ***<<*** *multiply*() ***<<*** endl;

      cout ***<<*** "Division is: " ***<<*** *divide*() ***<<*** endl;

    }

    T *add*(){

      return num1 + num2;

    }

    T *subtract*(){

      return num1 - num2;

    }

    T *multiply*(){

      return num1 \*  num2;

    }

    T *divide*(){

      return num1 / num2;

    }

};

int *main*(){

  Calculator<int>*intCalc*(2***,*** 1);

  Calculator<float>*floatCalc*(2.4***,*** 1.2);

  cout ***<<*** "Int results: " ***<<*** endl;

  intCalc***.****displayResult*();

  cout ***<<*** endl ***<<*** "Float results: " ***<<*** endl;

  floatCalc***.****displayResult*();

  return 0;

}

/\*

Int results:

Number are: 2 and 1.

Addition is: 3

Subtraction is: 1

Product is: 2

Division is: 2

Float results:

Number are: 2.4 and 1.2.

Addition is: 3.6

Subtraction is: 1.2

Product is: 2.88

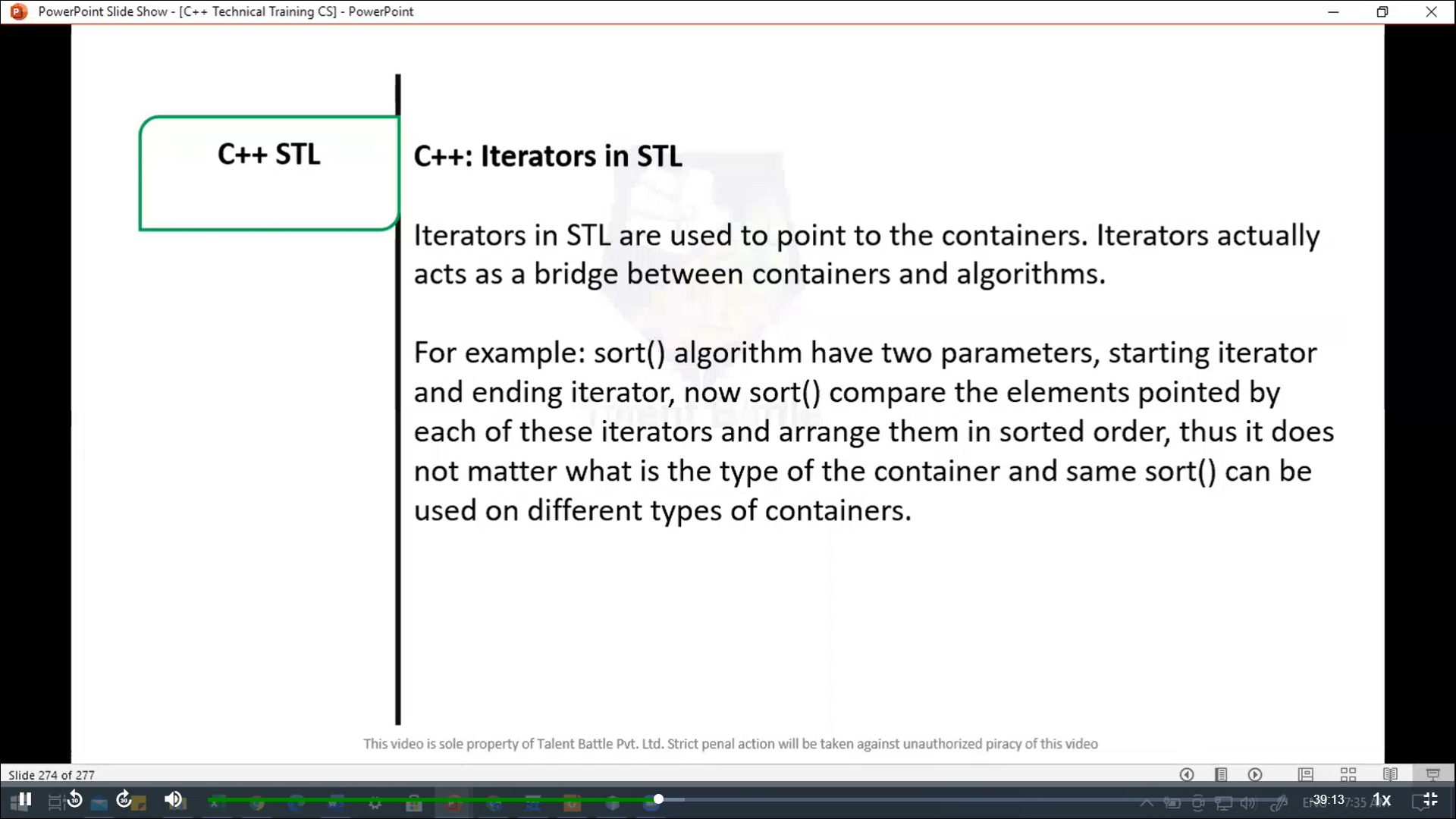
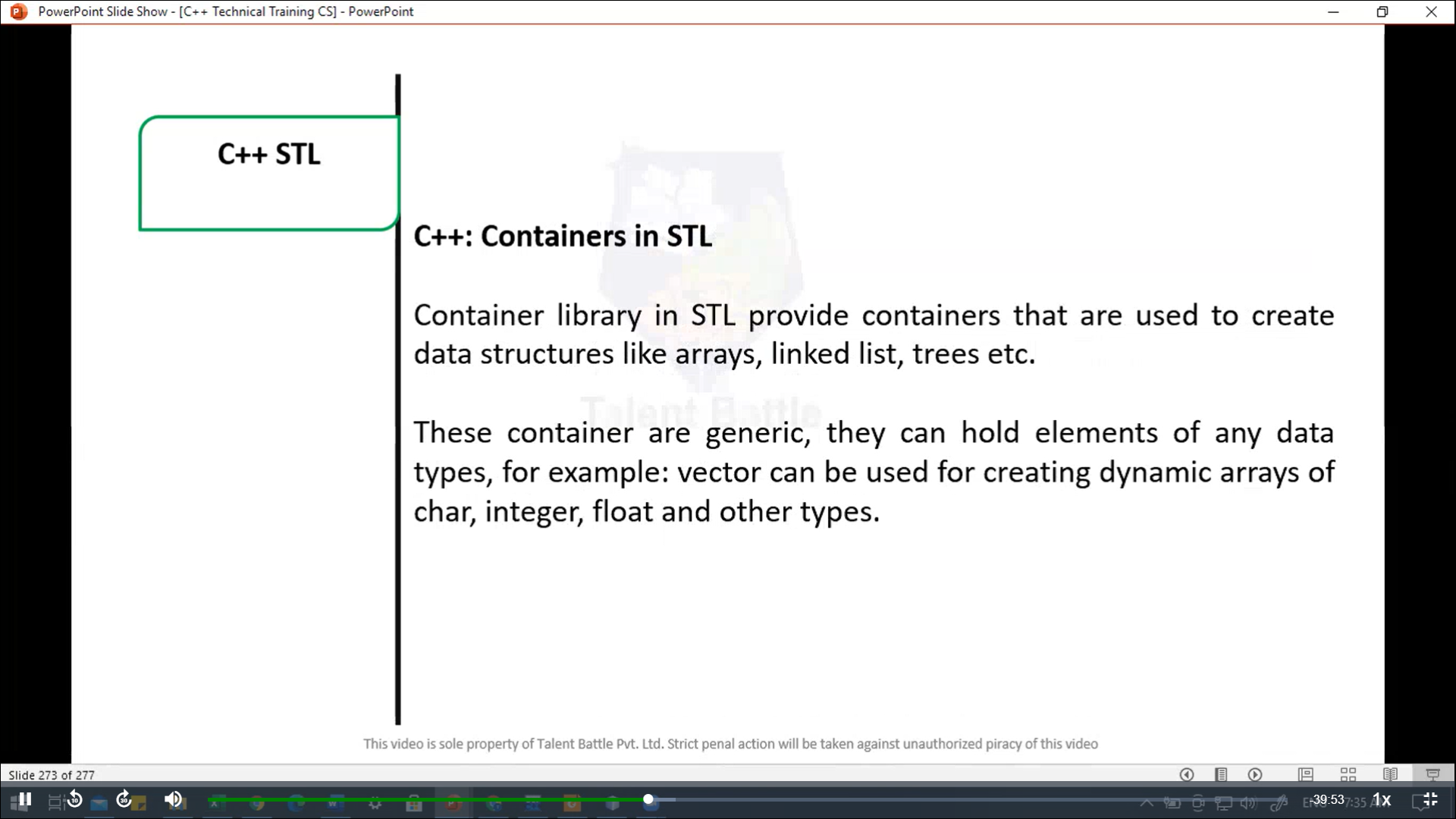
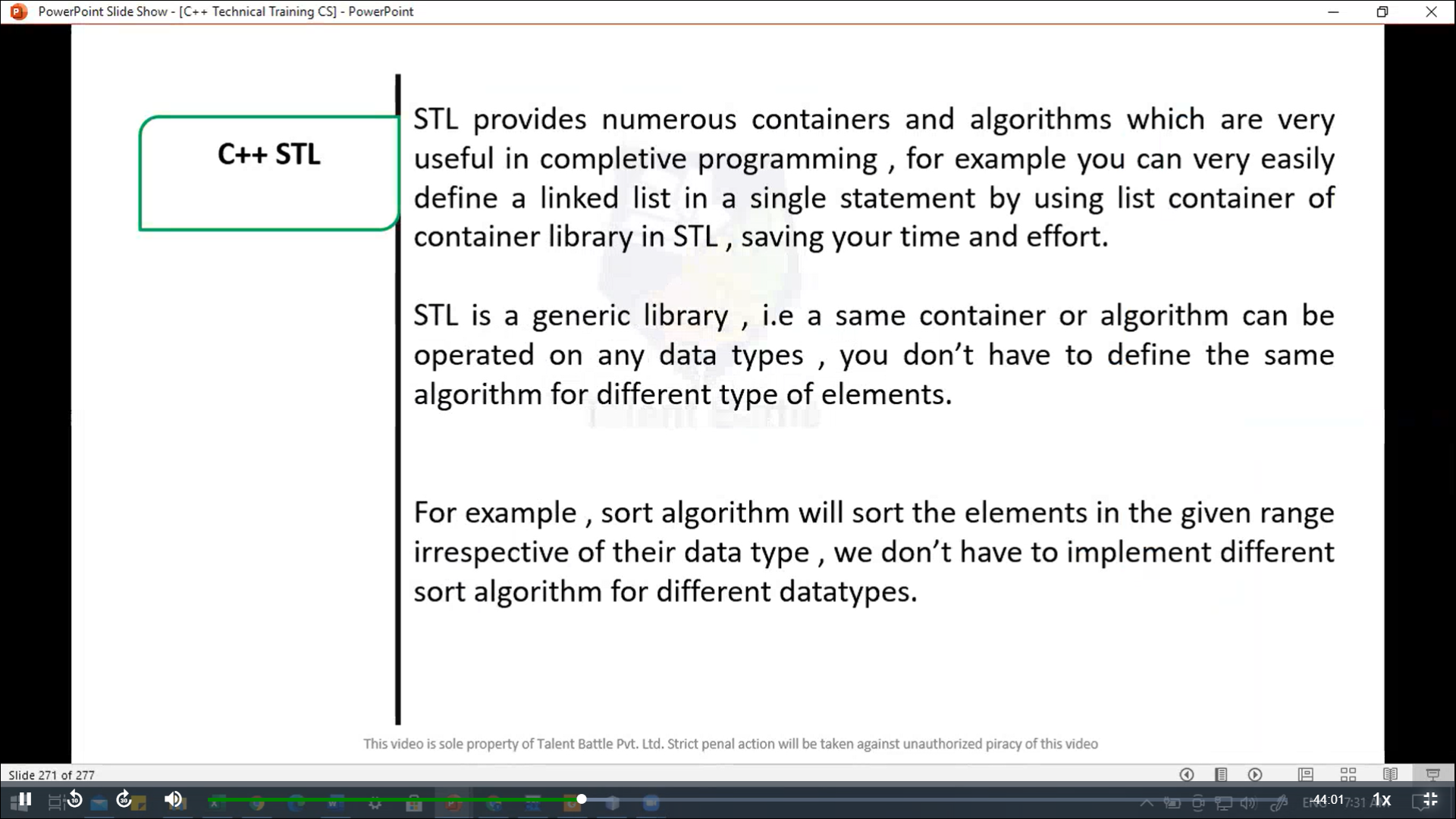
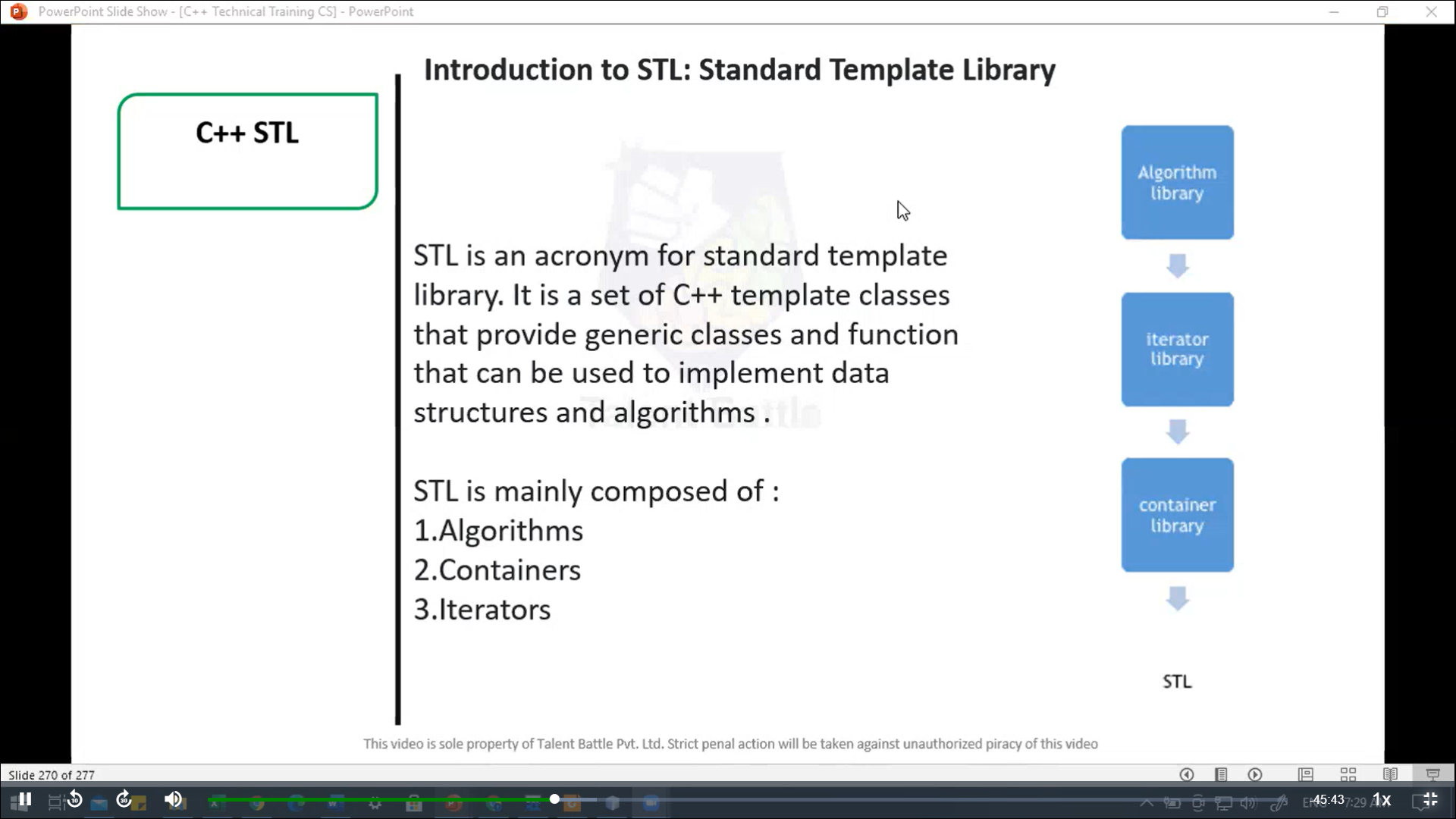
Division is: 2

--------------------------------

Process exited after 0.1501 seconds with return value 0

Press any key to continue . . .

\*/



//===================================================

// C++ STL

// C++ program to implement stack in stl

#include<iostream>

#include<stack>

#include<string>

#include<cstdlib>

using namespace ***std***;

int *main*(){

  stack<int>st;

  int choice***,*** item;

  while(1){

    cout  *<<* "**\n**-----------------------------"*<<endl*;

    cout *<<* "Stack implementation in Stl" *<<* *endl*;

    cout *<<* "**\n**-------------------------------"*<<* *endl*;

    cout *<<* "1. Insert Element into the Stack"*<<* *endl*;

    cout *<<* "2. Delete Element from the Stack" *<<* *endl*;

    cout *<<* "3. Size of the stack" *<<* *endl*;

    cout *<<* "4. Top Element of the Stack" *<<* *endl*;

    cout *<<* "5. Exit" *<<* *endl*;

    cout *<<* "Enter your Choice: ";

    cin *>>* choice;

***switch***(choice){

      case 1***:***

        cout *<<* "Enter value to be inserted: ";

        cin *>>* item;

        st***.****push*(item);

        break;

      case 2***:***

        item = st***.****top*();

        st***.****pop*();

        cout *<<* "Element " *<<* item *<<* " Deleted" *<<* *endl*;

        break;

      case 3***:***

        cout *<<* "Size of the Queue: ";

        cout *<<* st***.****size*() *<<* *endl*;

        break;

      case 4***:***

        cout *<<* "Top Element of the stack: ";

        cout *<<* st***.****top*() *<<* *endl*;

        break;

      case 5***:***

*exit*(1);

        break;

      default***:***

        cout *<<* "Wrong choice" *<<* *endl*;

    }

  }

  return 0;

}

/\*

-----------------------------

Stack implementation in Stl

-------------------------------

1. Insert Element into the Stack

2. Delete Element from the Stack

3. Size of the stack

4. Top Element of the Stack

5. Exit

Enter your Choice: 1

Enter value to be inserted: 55

-----------------------------

Stack implementation in Stl

-------------------------------

1. Insert Element into the Stack

2. Delete Element from the Stack

3. Size of the stack

4. Top Element of the Stack

5. Exit

Enter your Choice: 1

Enter value to be inserted: 33

-----------------------------

Stack implementation in Stl

-------------------------------

1. Insert Element into the Stack

2. Delete Element from the Stack

3. Size of the stack

4. Top Element of the Stack

5. Exit

Enter your Choice: 1

Enter value to be inserted: 33

-----------------------------

Stack implementation in Stl

-------------------------------

1. Insert Element into the Stack

2. Delete Element from the Stack

3. Size of the stack

4. Top Element of the Stack

5. Exit

Enter your Choice: 1

Enter value to be inserted: 22

-----------------------------

Stack implementation in Stl

-------------------------------

1. Insert Element into the Stack

2. Delete Element from the Stack

3. Size of the stack

4. Top Element of the Stack

5. Exit

Enter your Choice: 3

Size of the Queue: 4

-----------------------------

Stack implementation in Stl

-------------------------------

1. Insert Element into the Stack

2. Delete Element from the Stack

3. Size of the stack

4. Top Element of the Stack

5. Exit

Enter your Choice: 4

Top Element of the stack: 22

-----------------------------

Stack implementation in Stl

-------------------------------

1. Insert Element into the Stack

2. Delete Element from the Stack

3. Size of the stack

4. Top Element of the Stack

5. Exit

Enter your Choice: 1

Enter value to be inserted: 99

-----------------------------

Stack implementation in Stl

-------------------------------

1. Insert Element into the Stack

2. Delete Element from the Stack

3. Size of the stack

4. Top Element of the Stack

5. Exit

Enter your Choice: 4

Top Element of the stack: 99

-----------------------------

Stack implementation in Stl

-------------------------------

1. Insert Element into the Stack

2. Delete Element from the Stack

3. Size of the stack

4. Top Element of the Stack

5. Exit

Enter your Choice: 2

Element 99 Deleted

-----------------------------

Stack implementation in Stl

-------------------------------

1. Insert Element into the Stack

2. Delete Element from the Stack

3. Size of the stack

4. Top Element of the Stack

5. Exit

Enter your Choice: 5

--------------------------------

Process exited after 90.62 seconds with return value 1

Press any key to continue . . .

\*/