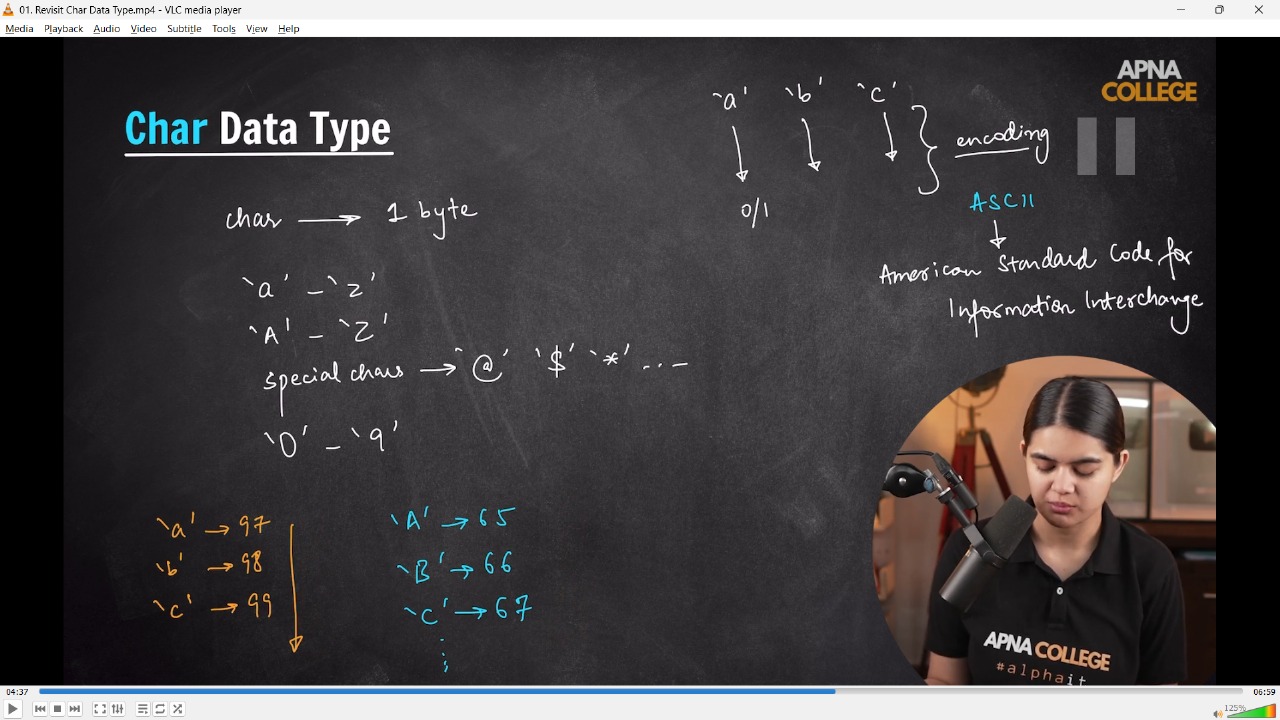
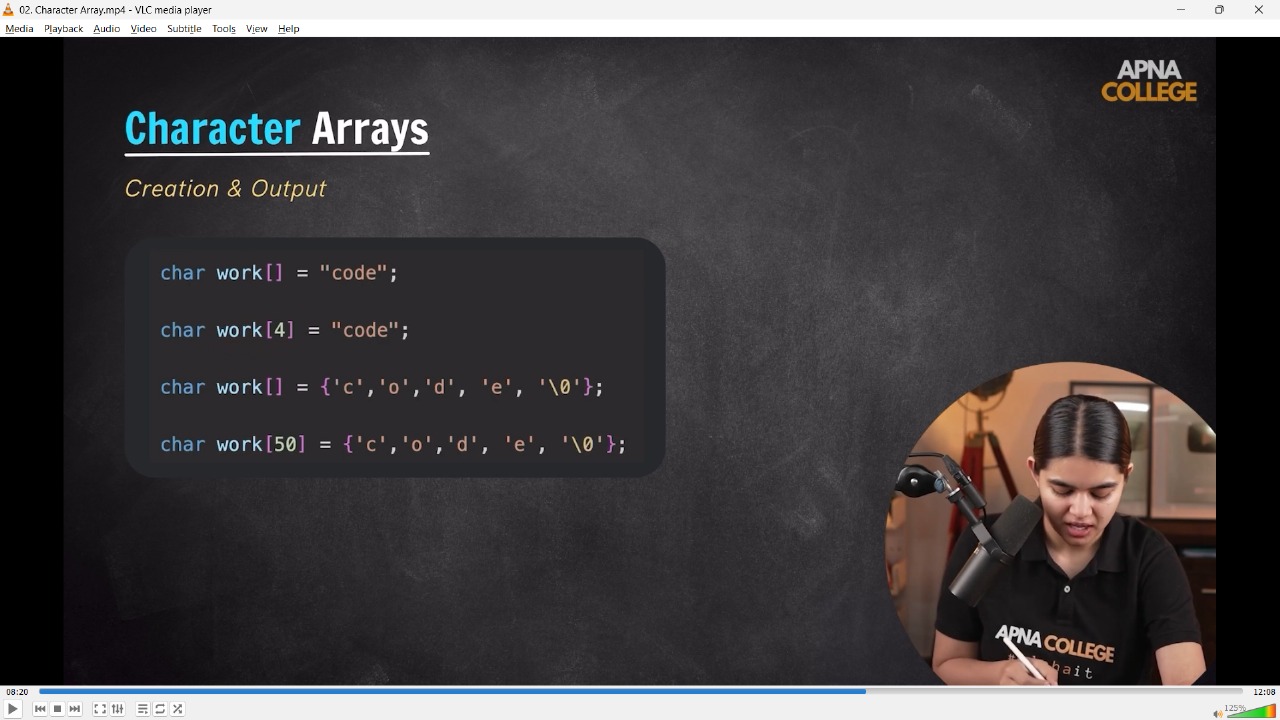
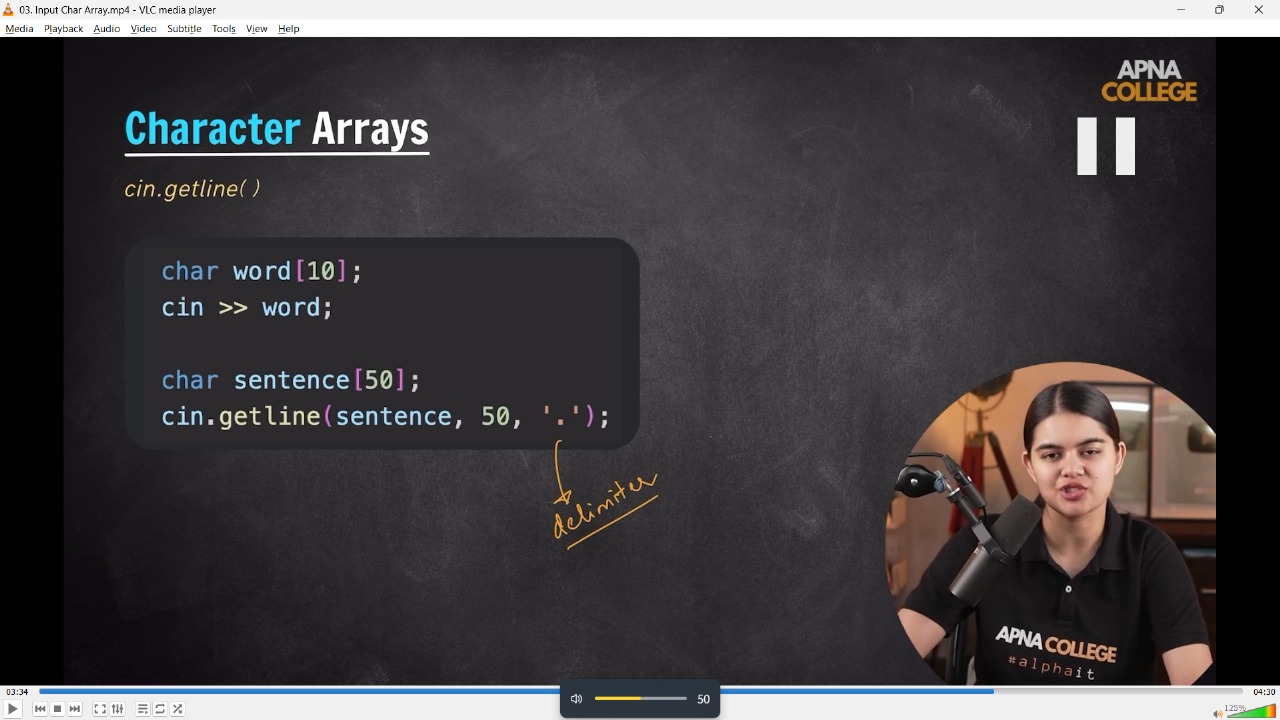
**5) Character Array & Strings in CPP –**

**// 1) Intro to Char Array –**







// int main()

// {

//     char ch = 'F';

//     int pos = ch - 'A';

//     cout << pos << endl; // 5

//     char arr[5] = {'a', 'b', 'c', 'd', 'e'};

//     cout << arr[0] << endl;

//     cout << arr[1] << endl;

//     cout << arr[2] << endl;

//     cout << arr[3] << endl;

//     cout << arr[4] << endl;

//     cout << arr[5] << endl;

//     char arr2[8] = {'s', 'h', 'u', 'b', 'h', 'a', 'm'};

//     cout << arr2 << endl; // shubham - in int array on printing array, it gives address of array in hexadecimal form, but in char array it directly gives the value

//     // at the end - /0 indicates that these are not letters only this is a valid string word.

//     char arr3[] = {'m', 'a', 'h', 'a', 'j', 'a', 'n', '\0'};

//     cout << arr3 << endl; // mahajan

//     "Microsoft"; // it's a string liiteral -literal means constant. which is a fix value

//     // String creation methods -

//     char arru[] = {'M', 'i', 'c', 'r', 'o', 's', 'o', 'f', 't', '\0'};

//     char arru2[12] = {'H', 'y', 'd', 'e', 'r', 'a', 'b', 'a', 'd'};

//     char work[] = "SoftwareDeveloper";

//     char office[] = "New Building";

//     cout << arru << endl;           // Microsoft

//     cout << work << endl;           // SoftwareDeveloper

//     cout << arru2 << endl;          // Hyderabad

//     cout << office << endl;         // NewBuilding

//     cout << strlen(work) << endl;   // 17

//     cout << strlen(office) << endl; // 12

//     // taking input on string  -

//     /\* char shabd[15];

//     cin >> shabd;

//     cout<<"So, you typed - "<<shabd<<"and the length is - "<<strlen(shabd)<<endl; \*/

//     /\*

//     Pune

//     So, you typed - Puneand the length is - 4

//      \*/

//     // for the input of full statement

//     /\* char carrier[50];

//     cin.getline(carrier, 50);

//     cout << "So, your entered string is - " << carrier << endl; // in Microsoft with the package of 51 LPA  \*/

//     // If i want to stop meanwhile the sentence

//     char carrier2[50];

//     cin.getline(carrier2, 50, '\*');

//     cout << "Entered String is - " << carrier2 << endl;

//     /\*

//     Only in Microsoft with \*  the package of 51 LPA

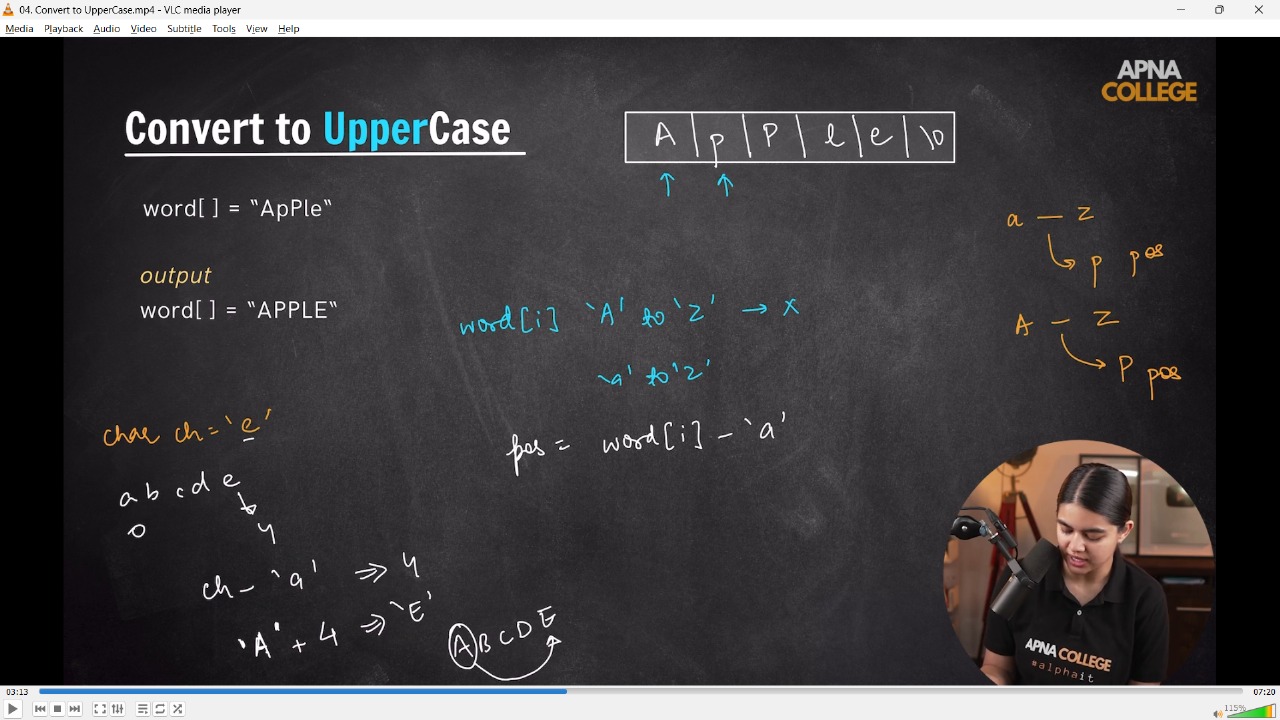
//     Entered String is - Only in Microsoft with

//      \*/

// }

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

**2) Covert to Upper Letter -**



// void toUpper(char word[], int n)

// {

//     for (int i = 0; i < n; i++)

//     {

//         char ch = word[i];

//         if (ch >= 'A' && ch <= 'Z') // if its already in Uppper Case

//         {

//             continue;

//         }

//         else

//         {

//             word[i] = ch - 'a' + 'A';

//         }

//     }

// }

// int main()

// {

//     cout << "What's in your mind - " << endl;

//     char word[50];

//     cin.getline(word, 50);

//     toUpper(word, strlen(word));

//     cout << word << endl;

//     /\*

//     What's in your mind -

//     MicrosftHyderabad

//     MICROSFTHYDERABAD

//      \*/

// }

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

**// 2.1) Covert to Lowercase Letetr -**

// void toUpper(char word[], int n)

// {

//     for (int i = 0; i < n; i++)

//     {

//         char ch = word[i];

//         if (ch >= 'a' && ch <= 'z') // if its already in Lower Case

//         {

//             continue;

//         }

//         else

//         {

//             word[i] = ch - 'A' + 'a';

//         }

//     }

// }

// int main()

// {

//     cout << "What's in your mind - " << endl;

//     char word[50];

//     cin.getline(word, 50);

//     toUpper(word, strlen(word));

//     cout << word << endl;

//     /\*

//     What's in your mind -

//     MicrosftHyderabad

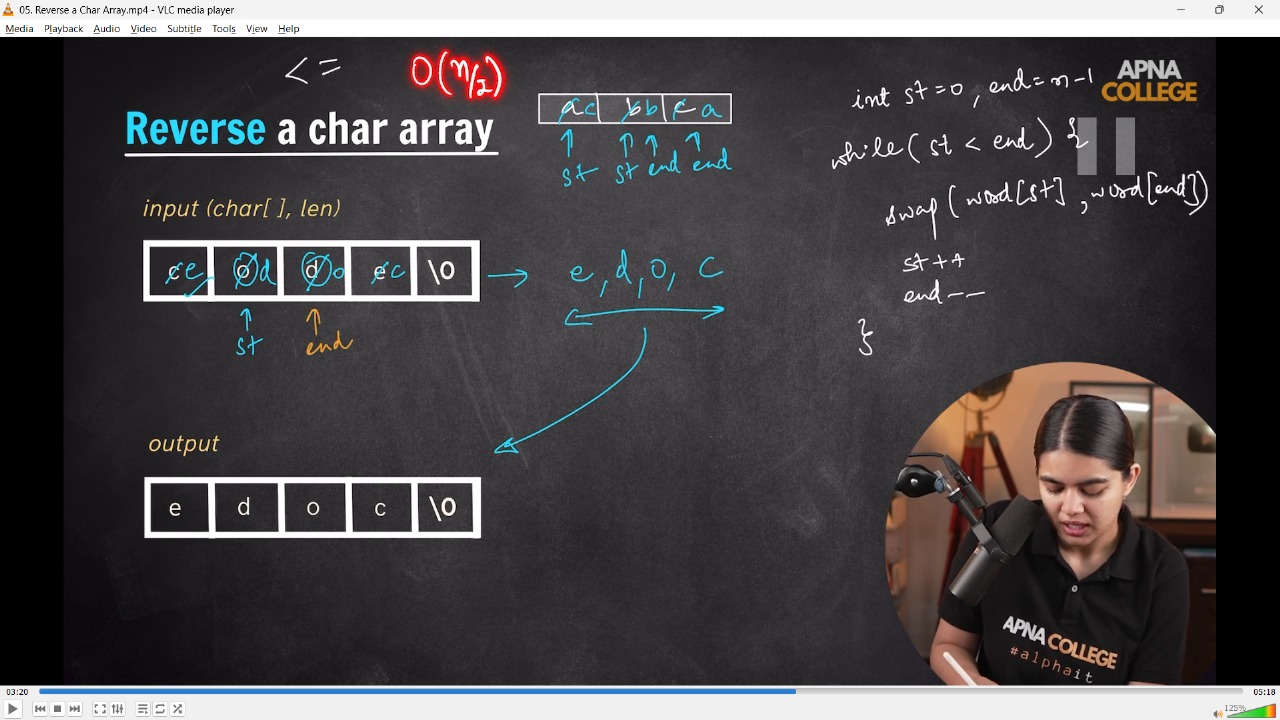
//     microsfthyderabad

//      \*/

// }

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

**3) Reverse a Char Array -**



// void ReverseCharArray(char word[], int n)

// {

//     int st = 0, end = n - 1;

//     while (st < end)

//     {

//         swap(word[st], word[end]);

//         st++;

//         end--;

//     }

//     cout << "Hence, the reversed word is - " << word << endl;

// }

// int main()

// {

//     cout << "What's in your mind - " << endl;

//     char word[50];

//     cin.getline(word, 50);

//     ReverseCharArray(word, strlen(word));

// /\*

// What's in your mind -

// MicrosoftHyderabad

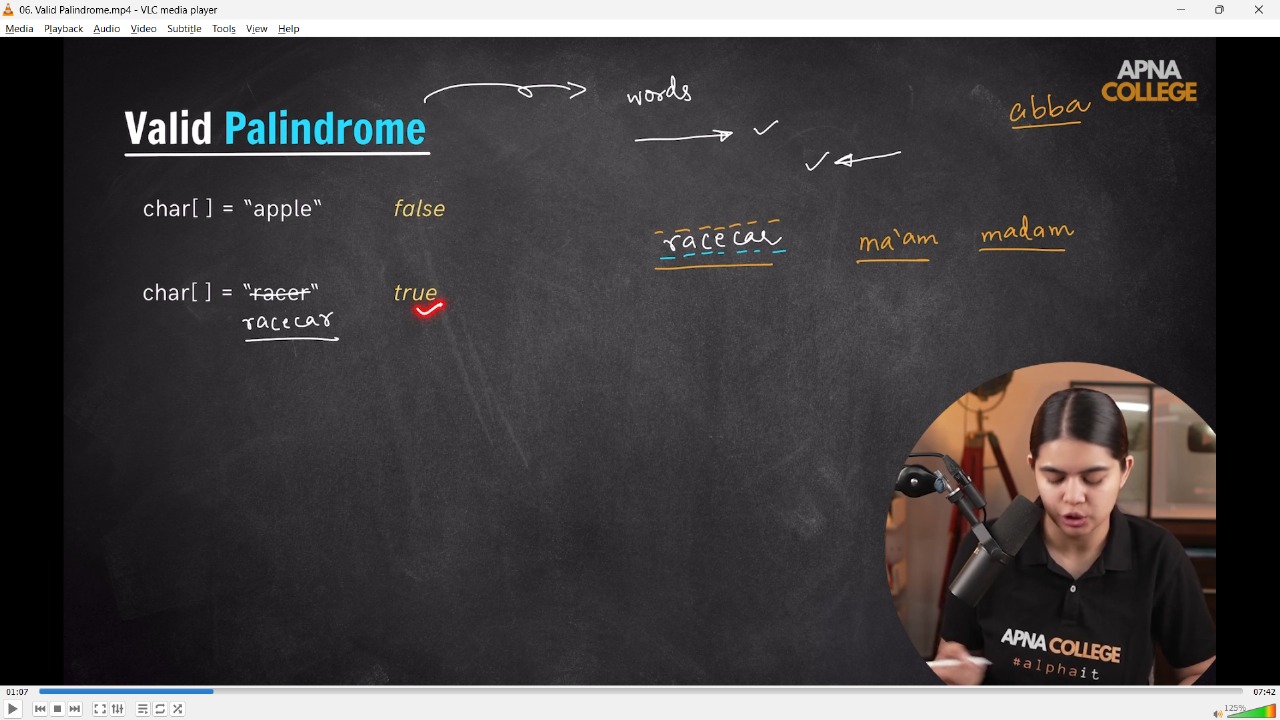
// Hence, the reversed word is - dabaredyHtfosorciM

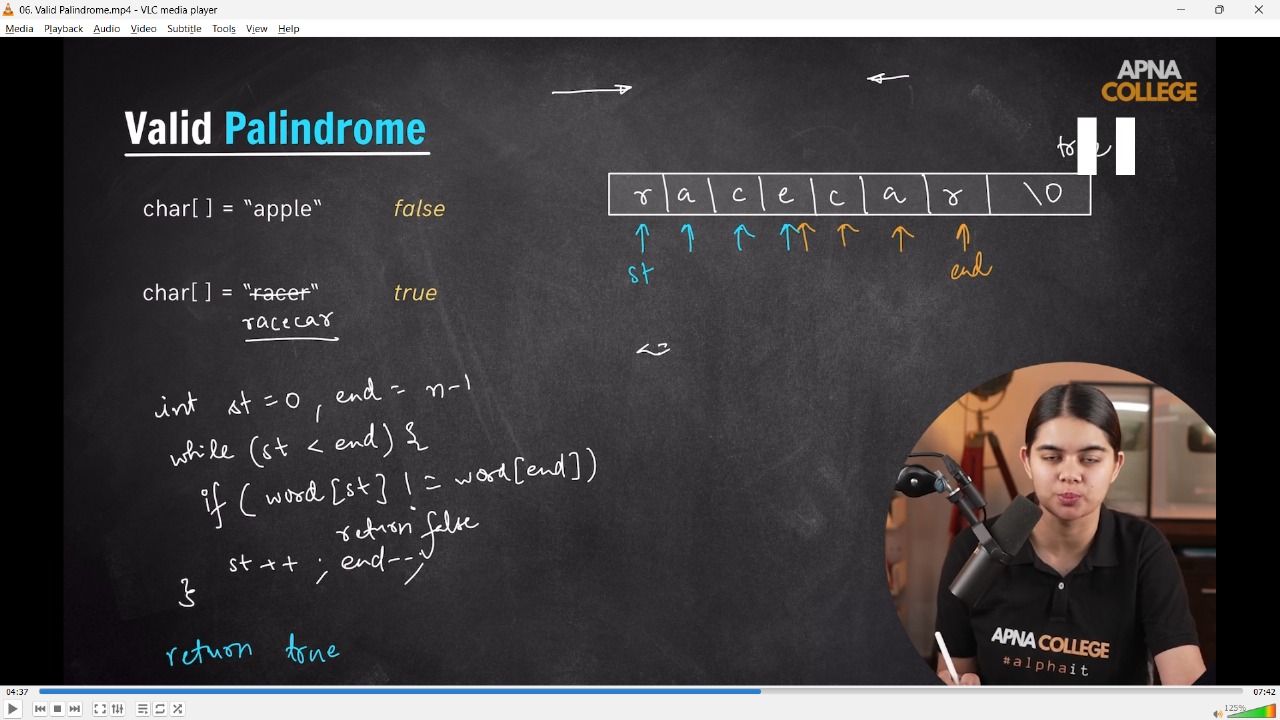
//  \*/

// }

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

**4) Valid Palindrome in char array -**





// bool PalindromeWord(char word[], int n)

// {

//     int st = 0, end = n - 1;

//     while (st < end)

//     {

//         if (word[st] != word[end])

//         {

//             cout << "Not valid Palindrome " << endl;

//             return false;

//         }

//         st++;

//         end--;

//     }

//     cout << "Valid Palindromed  " << endl;

//     return true;

// }

// int main()

// {

//     cout << "What's in your mind - " << endl;

//     char word[100];

//     cin.getline(word, 100);

//     PalindromeWord(word, strlen(word));

//     /\*

//     What's in your mind -

//     NaMaN

//     Valid Palindromed

//     What's in your mind -

//     naman

//     Valid Palindromed

//     What's in your mind -

//     Naman

//     Not valid Palindrome

//     What's in your mind -

//     racecar

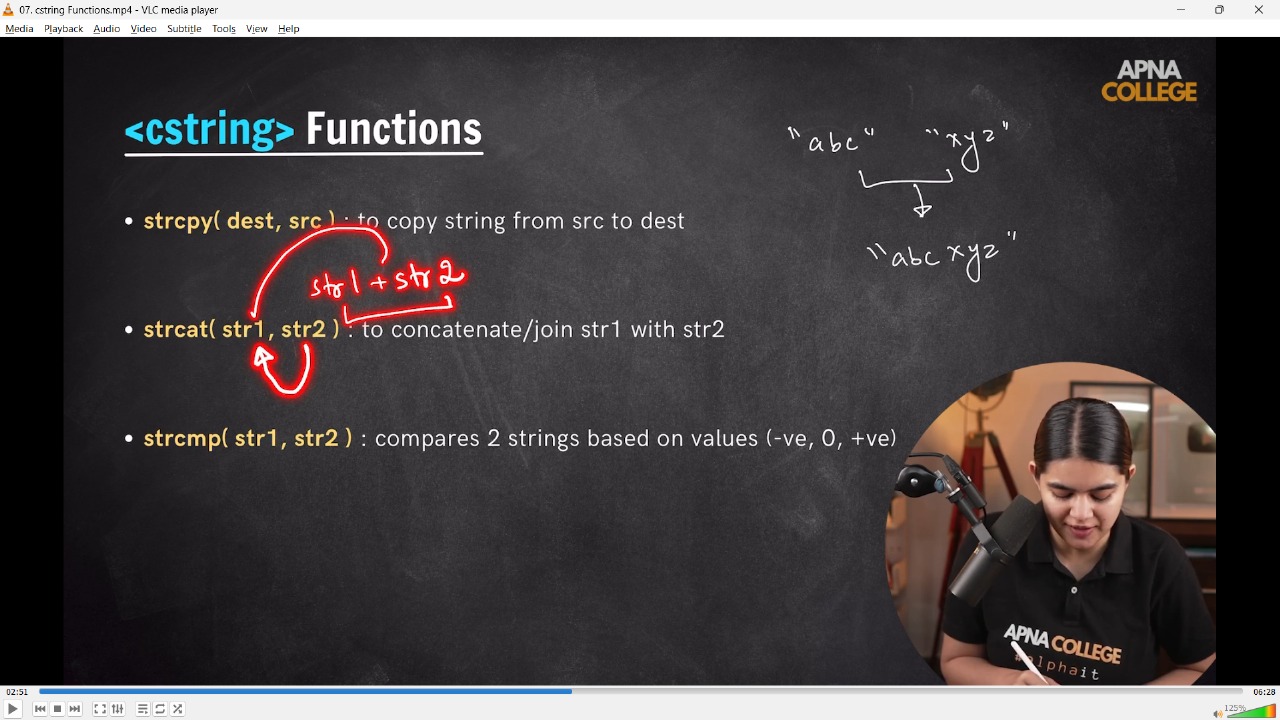
//     Valid Palindromed

//      \*/

// }

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

**5) Char Array(String) Functions -**



// int main()

// {

//     char word1[100];

//     // str1 = "Shubham Mahajan"; - as it's showing the error in assigning value, so for allowing it we can use cstring functions - that is strcpy

//     char word2[100] = "Microsoft Hyderabad";

//     strcpy(word1, "Free Offcie tour");

//     cout << word1 << endl; // Free Offcie tour

//     // For String Concatenation -

//     char word3[100] = "Let's come inside";

//     strcat(word2, word3);

//     cout << word2 << endl; // Microsoft HyderabadLet's come inside

//     // For String Comparision -

//     char word4[] = "Microsoft Hyderabad";

//     cout << strcmp(word2, word4) << endl; // 1

//     char word5[100] = "abc";

//     char word6[100] = "xyz";

//     cout<<strcmp(word5,word6)<<endl;//-1 - if the baad waali string is big, then returns negtive value

//     char word7[100] = "xyz";

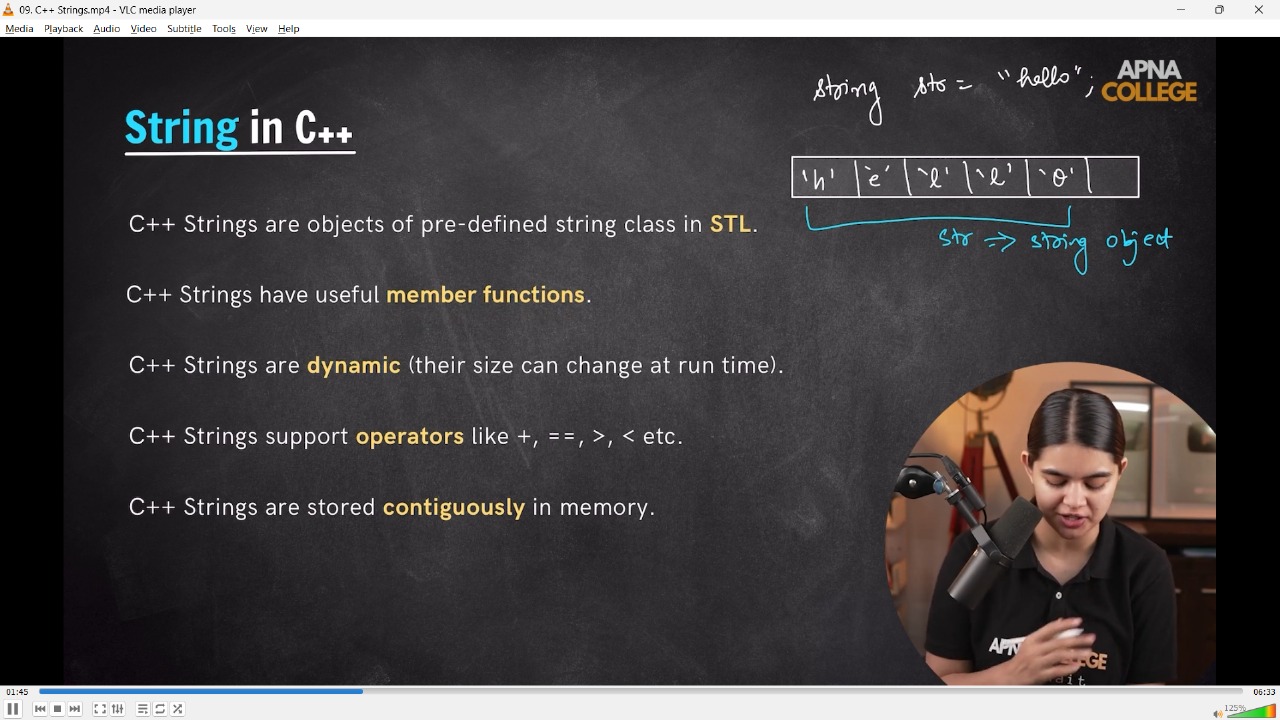
//     char word8[100] = "abc";

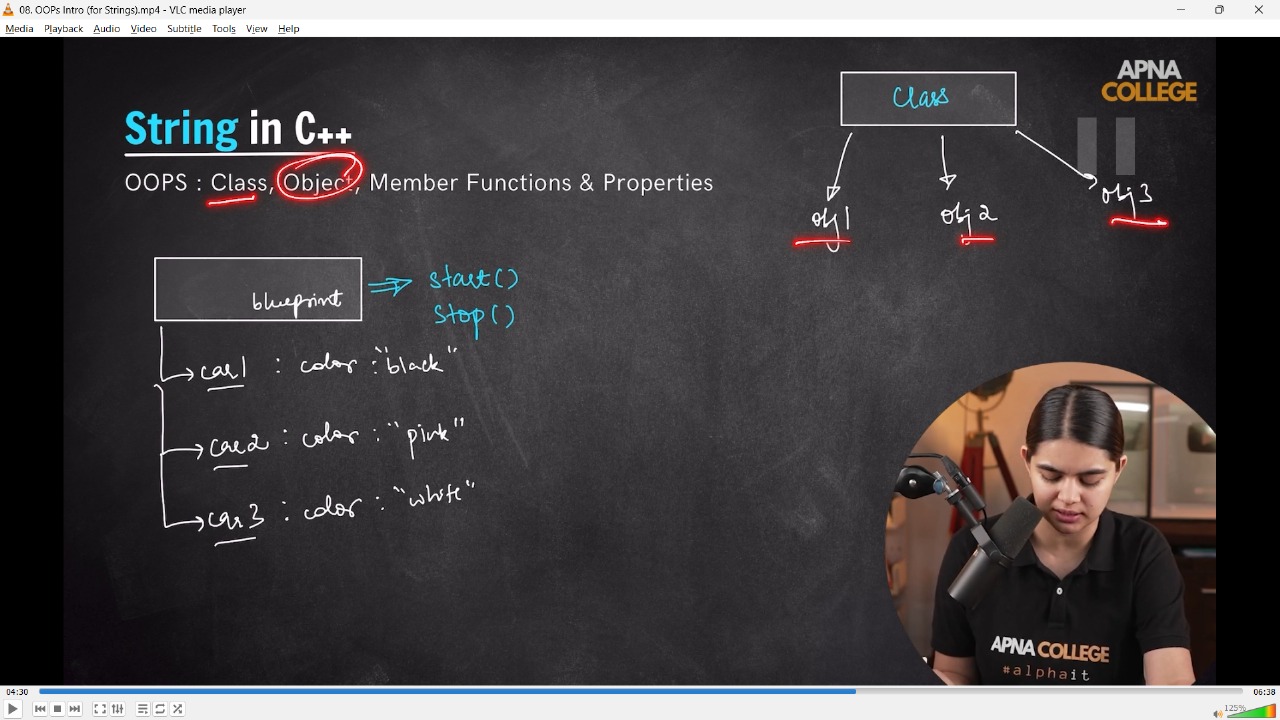
//     cout<<strcmp(word7,word8)<<endl;//1 - if the phle waali string is big, then returns possitive value

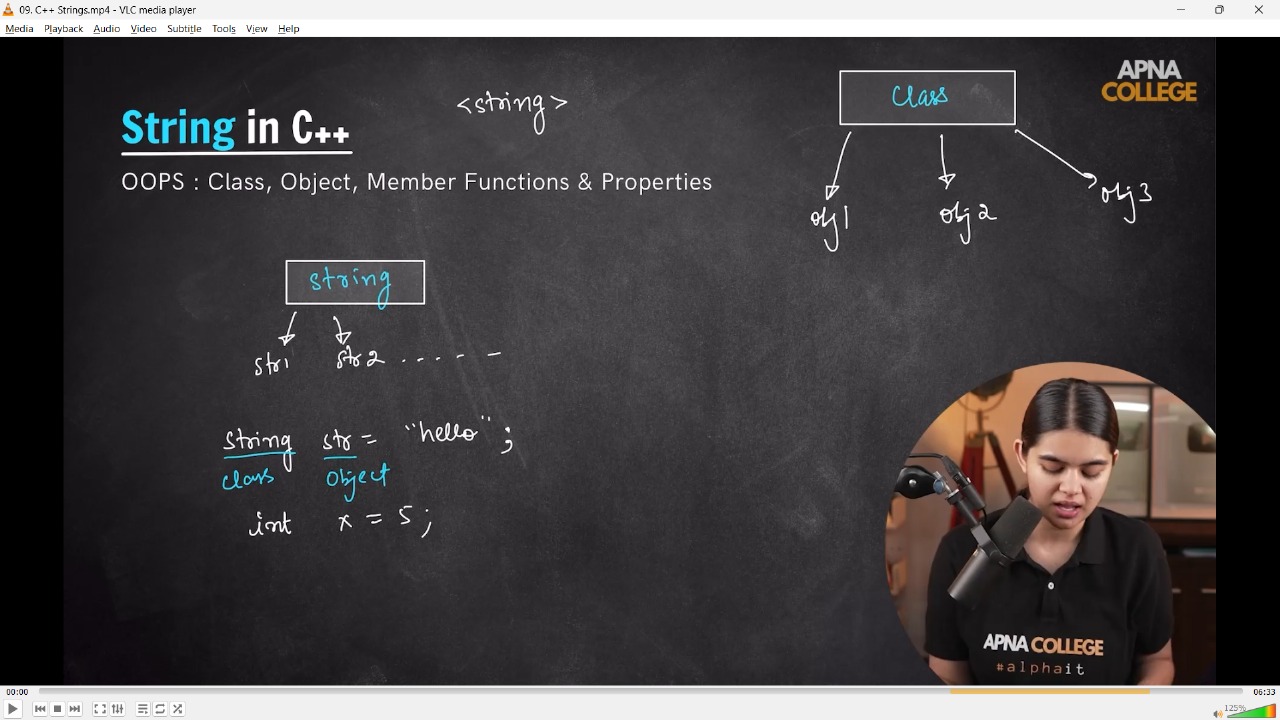
// }

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

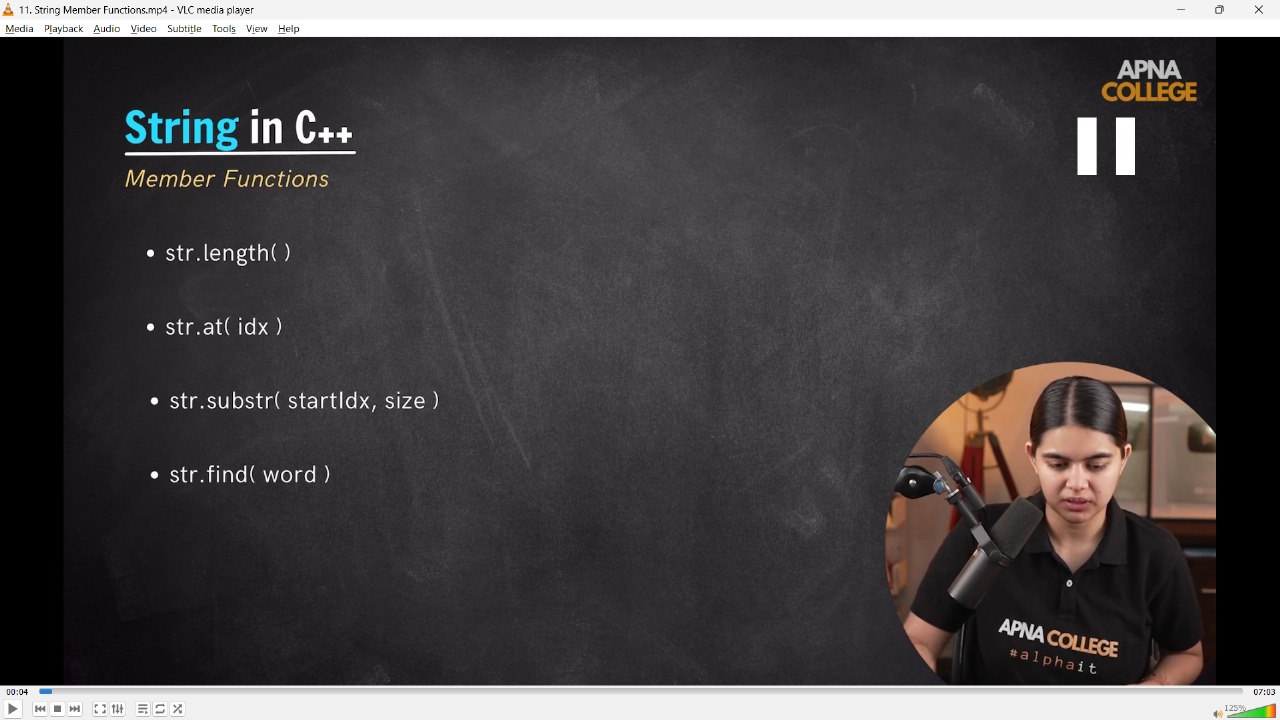
**6)String Intro & Basic I/p, o/p operations –**

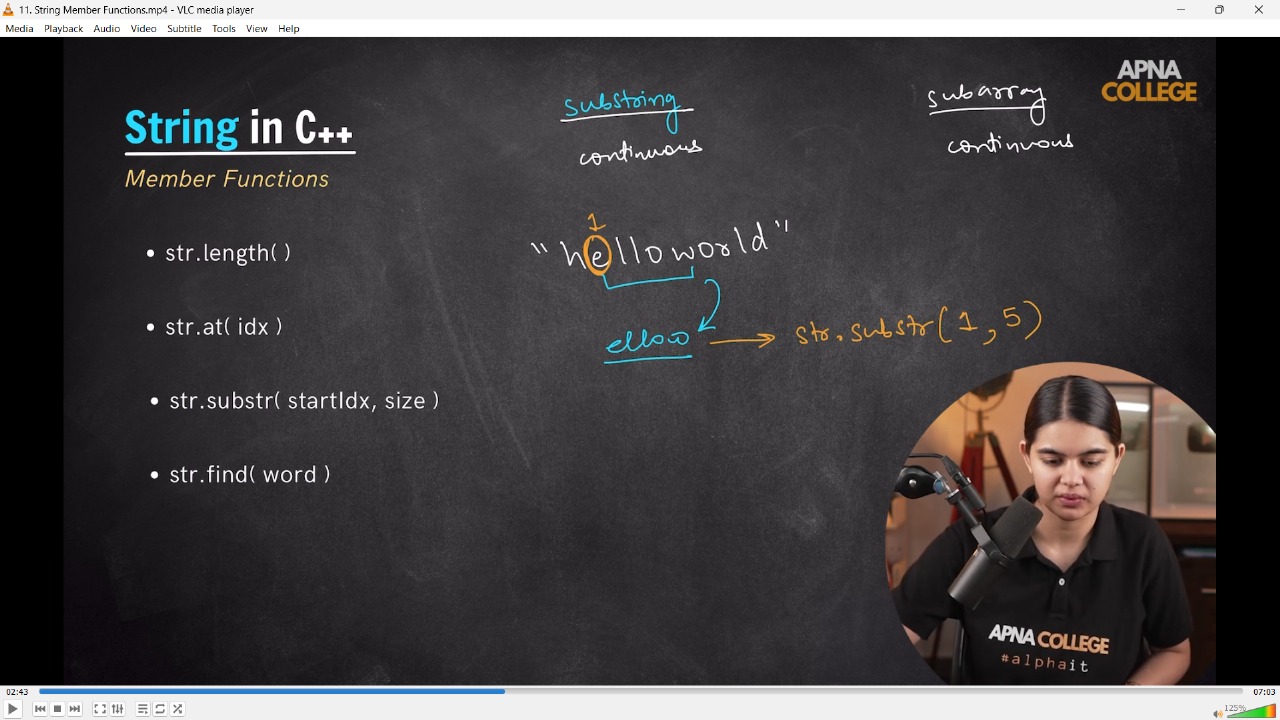


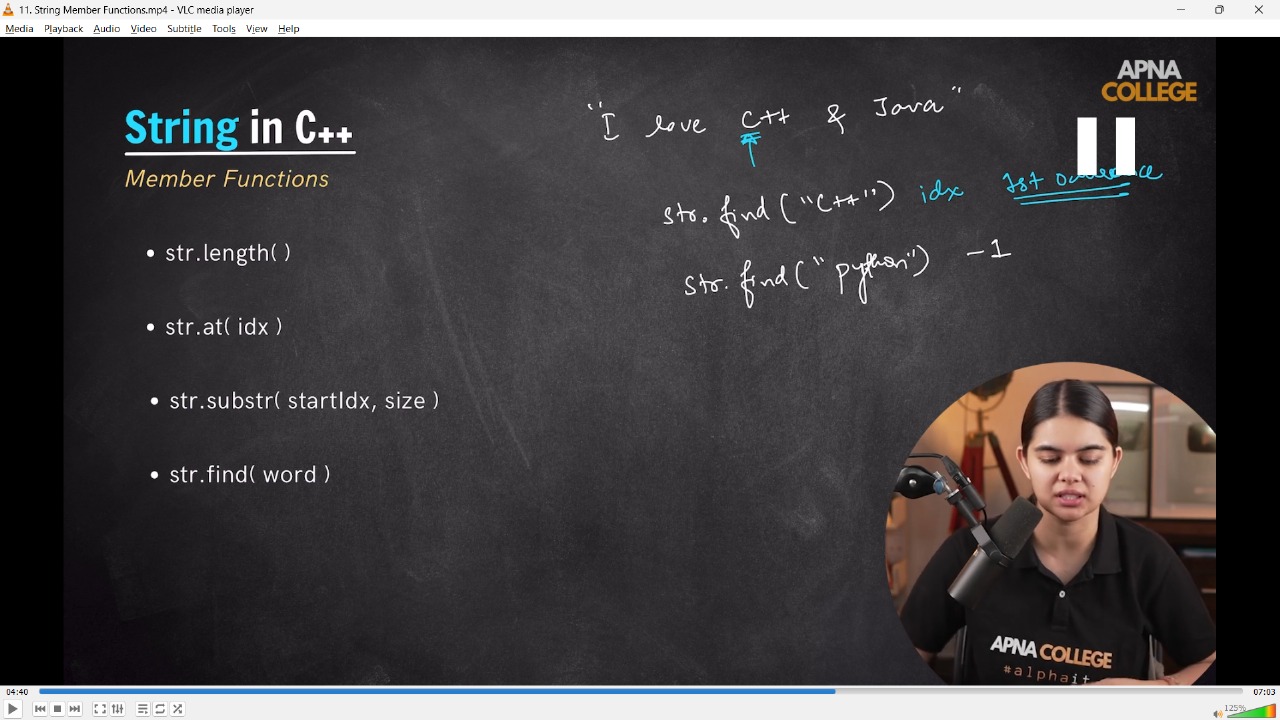




**6.1) String Memeber Functions-**







// int main()

// {

//     string str = "Microsoft Hyderabad";

//     cout << str.length() << endl;      // 19

//     cout << str[3] << endl;            // r

//     cout << str.at(3) << endl;         // r

//     cout << str.substr(5, 17) << endl; // soft Hyderabad

//     string str2;

//     cout << "what's in your mind baby - " << endl;

//     getline(cin, str2);                     // Microsoft's new office in Pune

//     cout << str2.find("Pune") << endl;      // 26

//     cout << str2.find("GIFT CITY") << endl; // 18446744083781055516154 - which is a Garbage value and eq. to -1. FInd fun returns a unsigned value which is this.

//     // want to see this as -1

//     int idx = str2.find("GIFT CITY");

//     cout << idx << endl; //-1

//     // Simialrlty for more than one occurence. Just start indexing before that occurence to count that

//     string str3;

//     cout << "what's in your mind baby - " << endl;

//     getline(cin, str3);

//     cout << str3.find("office") << endl;     // 26

//     cout << str3.find("office", 30) << endl; // 53

//     cout << str3.find("office", 54) << endl; // 96

//     // String Operations -

//     string t1 = "cat";

//     string t2 = "dog";

//     string t1 = "cat";

//     string t2 = "dog";

//     cout << (t1 != t2) << endl;//1

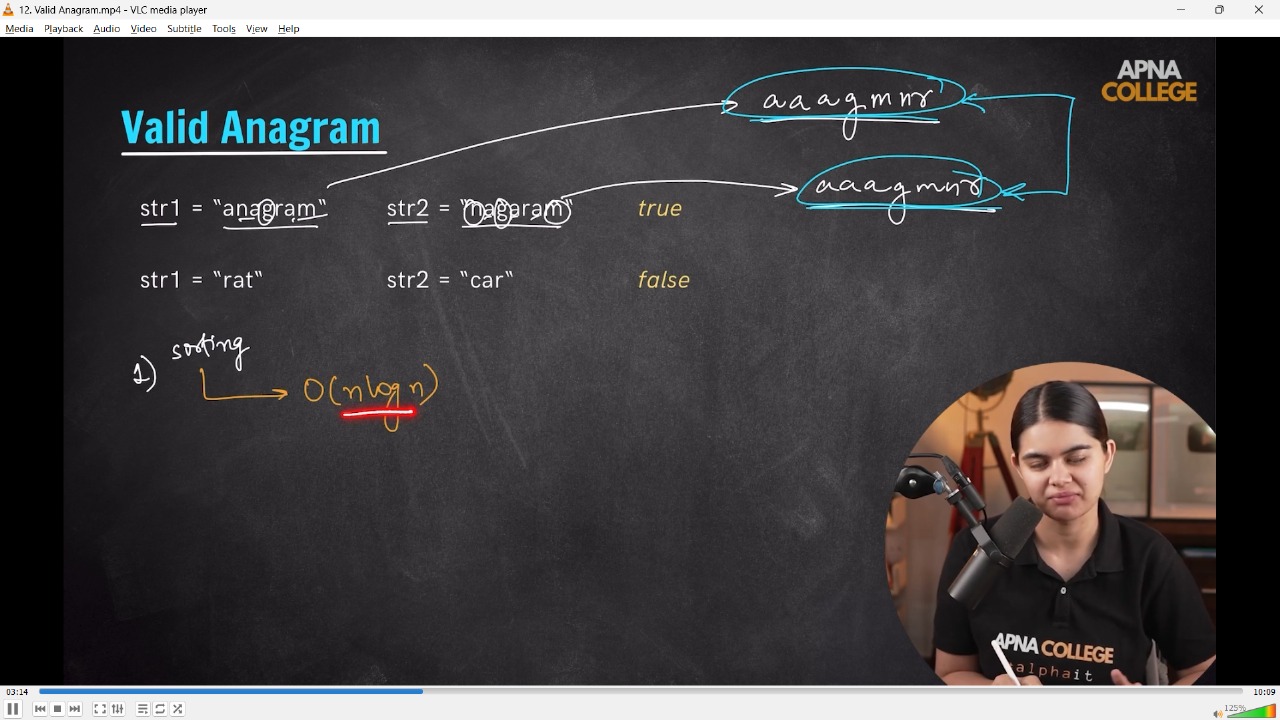
//     cout << (t1 > t2) << endl;//0

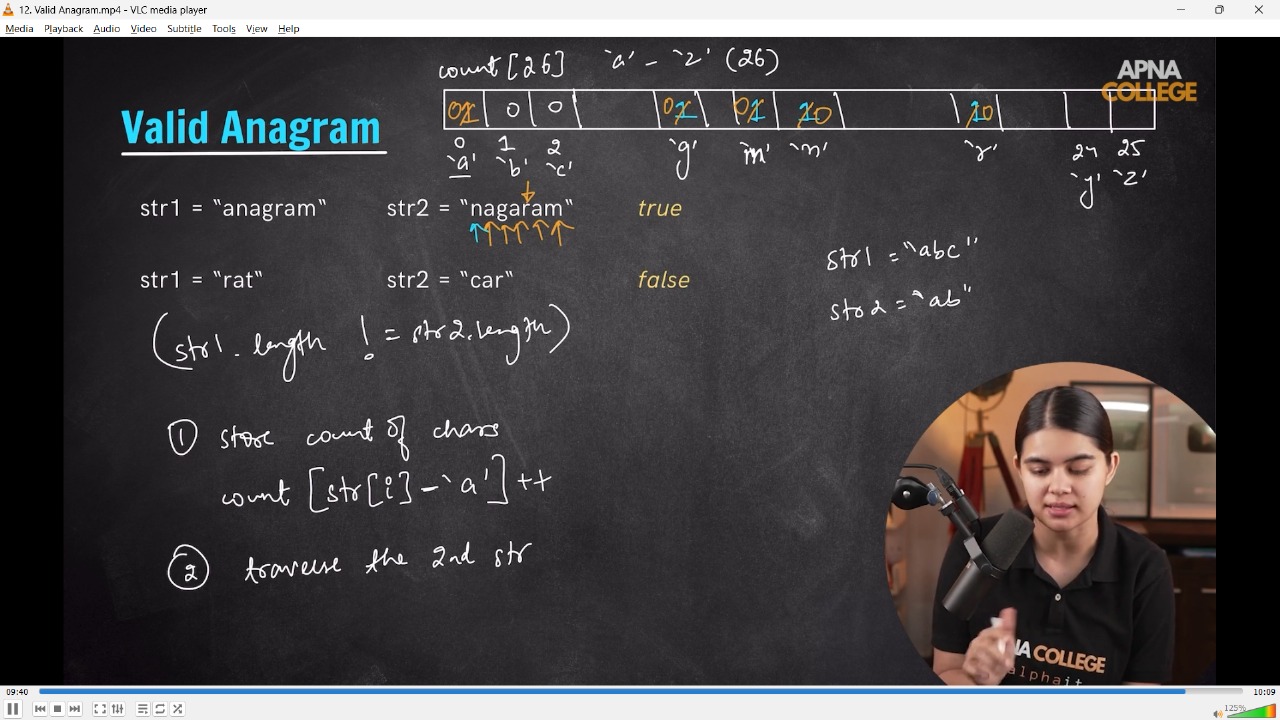
//     cout << (t1 < t2) << endl;//1

// }

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

**7) Check the two strings are Anagram or not –**





// bool isAnagram(string str1, string str2)

// {

//     if (str1.size() != str2.size())

//     {

//         cout << "not valid ANagrams" << endl;

//         return false;

//     }

//     int count[26] = {0}; // setting all character's count 0 initially

//     for (int i = 0; i < str1.length(); i++)

//     {

//         count[str1[i] - 'a']++;

//     }

//     for (int i = 0; i < str2.length(); i++)

//     {

//         if (count[str2[i] - 'a'] == 0)

//         {

//             cout << "It's not a valid ANagam" << endl;

//             return false;

//         }

//         count[str2[i] - 'a']--;

//     }

//     cout << "Valid ANagrams" << endl;

//     return true;

// }

// int main()

// {

//     string str1, str2;

//     cout << "value of string 1 - " << endl;

//     getline(cin, str1);

//     cout << "Similarly, value of string 2 is - " << endl;

//     getline(cin, str2);

//     isAnagram(str1, str2);

//     /\*

//     value of string 1 -

//     anagram

//     Similarly, value of string 2 is -

//     nagaram

//     Valid ANagrams

//     value of string 1 -

//     abcdef

//     Similarly, value of string 2 is -

//     dabfec

//     Valid ANagrams

// value of string 1 -

// shruti

// Similarly, value of string 2 is -

// rutish

// Valid ANagrams

// value of string 1 -

// madhuri

// Similarly, value of string 2 is -

// ridhum

// not valid ANagrams

//     T.C - O(n+m) or any of them which dominates as per the length of string 1 or 2

//     \*/

// }

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