

Day 3

C++ Recursion, Arrays, Strings, Structures

PowerPoint Slide Show - [C++ Technical Training CS] - PowerPoint

C++ Functions

C++ Recursion

A function that calls itself is known as a recursive function. And, this technique is known as recursion.

Working of Recursion in C++

```
void recurse()
{
    ...
    recurse();
    ...
}

int main()
{
    ...
    recurse();
    ...
}
```

This video is sole property of Talent Battle Pvt. Ltd. Strict penal action will be taken against unauthorized piracy of this video

Slide 119 of 173

PowerPoint Slide Show - [C++ Technical Training CS] - PowerPoint

C++ Functions

```
void recurse() {
    ...
    recurse();    ← recursive call
    ...
}

int main() {
    ...
    recurse();    ← function call
    ...
}
```

This video is sole property of Talent Battle Pvt. Ltd. Strict penal action will be taken against unauthorized piracy of this video

Slide 120 of 173

```
// C++ Recursion, Arrays, Strings, Structures

// Factorial of n = 1*2*3*...*n
#include <iostream>
using namespace std;

int factorial(int); // function prototype

int main(){
    int n, result;
    cout << "Enter a non-negative number: ";
    cin >> n;
    result = factorial(n);
    cout << "Factorial of " << n << " = " << result;
    return 0;
}

int factorial(int n) {
    if(n > 1){
        cout << n << " " << n*factorial(n-1) << endl;
        return n * factorial(n-1);
    } else {
        return 1;
    }
}
```

=====

Output:

Enter a non-negative number: 5

Factorial of 5 = 120

Process exited after 2.145 seconds with return value 0

C++ Functions

```
// Factorial of n = 1*2*3*...*n

#include <iostream>
using namespace std;
int factorial(int);
int main() {
    int n, result;
    cout << "Enter a non-negative number: ";
    cin >> n;
    result = factorial(n);
    cout << "Factorial of " << n << " = " << result;
    return 0;
}
int factorial(int n) {
    if (n > 1) {
        return n * factorial(n - 1);
    } else {
        return 1;
    }
}
```

Handwritten annotations:

- A red arrow points from the word "prototype" to the declaration of the function "factorial(int)".
- The variable "n" is circled and labeled "n = 5".
- The variable "result" is circled and labeled "result = factorial(5)".
- The expression "5 * 4 * 3 * 2 * 1" is written below the factorial call.
- A vertical stack of numbers 5, 4, 3, 2, 1 is shown, with arrows indicating they are being multiplied together. To the right of this stack, the recursive calls are listed: "5 * factorial(4)", "4 * factorial(3)", "3 * factorial(2)", "2 * factorial(1)", and "1".

This video is sole property of Talent Battle Pvt. Ltd. Strict penal action will be taken against unauthorized piracy of this video.

C++ Functions

Advantages of C++ Recursion

- It makes our code shorter and cleaner.
- Recursion is required in problems concerning data structures and advanced algorithms, such as Graph and Tree Traversal.

Disadvantages of C++ Recursion

- It takes a lot of stack space compared to an iterative program.
- It uses more processor time.
- It can be more difficult to debug compared to an equivalent iterative program.

This video is sole property of Talent Battle Pvt. Ltd. Strict penal action will be taken against unauthorized piracy of this video.

```
//=====
// C++ Return by Reference
/*
In C++ Programming, not only can you pass values by reference
to a function but you can also return a value by reference.
*/

#include <iostream>
using namespace std;

// Global variable
int num;

// function declaration
int& test();

int main(){
    test() = 5;
    cout << num;
    return 0;
}

int& test(){
    return num;
}
```

```
=====
```

5

Process exited after 0.1432 seconds with return value 0

PowerPoint Slide Show - [C++ Technical Training CS] - PowerPoint

C++ Return by Reference

C++ Functions

In C++ Programming, not only can you pass values by reference to a **function** but you can also return a value by reference.

```
#include <iostream>
using namespace std;
// Global variable
int num;
// Function declaration
int& test(); prototype
int main()
{
    test() = 5;
    cout << num;
    return 0;
}
int& test()
{
    return num;
}
```

int test();

5

Address of 5?

This video is sole property of Talent Battle Pvt. Ltd. Strict penal action will be taken against unauthorized piracy of this video

Slide 123 of 173

PowerPoint Slide Show - [C++ Technical Training CS] - PowerPoint

C++ Functions

Important Things to Remember When Returning by Reference.

Ordinary function returns value but this function doesn't. Hence, you cannot return a constant from the function.

```
int& test() {
    return 2;
}
```

You cannot return a local variable from this function.

```
int& test()
{
    int n = 2;
    return n;
}
```

This video is sole property of Talent Battle Pvt. Ltd. Strict penal action will be taken against unauthorized piracy of this video

Slide 124 of 173

PowerPoint Slide Show - [C++ Technical Training CS] - PowerPoint

C++ Functions

C++ Program to Display Prime Numbers Between Two Intervals Using Functions
C++ Program to Check Prime Number By Creating a Function
C++ Program to Check Whether a Number can be Express as Sum of Two Prime Numbers
C++ program to Find Sum of Natural Numbers using Recursion
C++ program to Calculate Factorial of a Number Using Recursion
C++ Program to Find G.C.D Using Recursion
C++ Program to Convert Binary Number to Decimal and vice-versa
C++ Program to Convert Octal Number to Decimal and vice-versa
C++ Program to Convert Binary Number to Octal and vice-versa
C++ program to Reverse a Sentence Using Recursion
C++ Program to Calculate Power Using Recursion

This video is sole property of Talent Battle Pvt. Ltd. Strict penal action will be taken against unauthorized piracy of this video

Slide 125 of 173

PowerPoint Slide Show - [C++ Technical Training CS] - PowerPoint

C++ Arrays

In C++, an array is a variable that can store multiple values of the same type.

In C++, the size and type of arrays cannot be changed after its declaration.

C++ Array Declaration
dataType arrayName[arraySize];
For example,

```
int x[6];
```

Here,

int - type of element to be stored
x - name of the array
6 - size of the array

This video is sole property of Talent Battle Pvt. Ltd. Strict penal action will be taken against unauthorized piracy of this video

```
//=====
// C++ Program to display prime number between two intervals using functions

#include <iostream>
using namespace std;

// function to check if a number is prime
bool isPrime(int num){
    if(num <= 1) // Prime numbers are greater than 1
        return false;

    for(int i = 2; i <= num / 2; ++i){
        if(num % 2 == 0)
            return false;
    }
    return true;
}

// function to print prime numbers between two intervals
void printPrimeInRange(int start, int end){
    cout << "Prime number between " << start << " and " << end << " are: ";
    for(int i = start; i <= end; ++i){
        if(isPrime(i)){
            cout << i << " ";
        }
    }
    cout << endl;
}

int main(){
    int start, end;

    // input from the user
    cout << "Enter the starting number: ";
    cin >> start;
    cout << "Enter the ending number: ";
    cin >> end;

    // call function to display prime numbers between the range
    printPrimeInRange(start, end);

    return 0;
}
/*
Enter the starting number: 1
Enter the ending number: 15
Prime number between 1 and 15 are: 2 3 5 7 9 11 13 15

-----
Process exited after 9.728 seconds with return value 0
```

```
*/
```

```
//=====
// C++ program to check prime number by creating a function
#include<iostream>
using namespace std;

// function to check if a number is prime
bool isPrime(int num){
    if(num <= 1) // prime number are greater than 1
    {
        return false;
    }

    for(int i = 2; i <= num/2; ++i){
        if(num % i == 0) // if the number is divisible by any number other than 1 and itself
        {
            return false;
        }
    }
    return true; // If no divisors found, it is prime
}

int main(){
    int number;

    // Input from the user
    cout << "Enter a number: ";
    cin >> number;

    // check if the number is prime using the function
    if(isPrime(number)){
        cout << number << " is a prime number." << endl;
    } else {
        cout << number << " is not a prime number." << endl;
    }

    return 0;
}
```

C++ Arrays

Access Elements in C++ Array

In C++, each element in an array is associated with a number. The number is known as an array index. We can access elements of an array by using those indices.

```
// syntax to access array elements
array[index];
```

Consider the array `x` we have seen above.



This video is sole property of Talent Battle Pvt. Ltd. Strict penal action will be taken against unauthorized piracy of this video

Slide 127 of 173



C++ Arrays

Few Things to Remember:

The array indices start with 0. Meaning `x[0]` is the first element stored at index 0.

If the size of an array is n , the last element is stored at index $(n-1)$. In this example, `x[5]` is the last element.

Elements of an array have consecutive addresses. For example, suppose the starting address of `x[0]` is 2120d. Then, the address of the next element `x[1]` will be 2124d, the address of `x[2]` will be 2128d and so on.

Here, the size of each element is increased by 4. This is because the size of int is 4 bytes.

This video is sole property of Talent Battle Pvt. Ltd. Strict penal action will be taken against unauthorized piracy of this video

Slide 128 of 173



C++ Arrays

C++ Multidimensional Arrays

In C++, we can create an array of an array, known as a multidimensional array. For example:

```
int x[3][4];
```

Here, x is a two-dimensional array. It can hold a maximum of 12 elements. We can think of this array as a table with 3 rows and each row has 4 columns as shown below.

| | Col 1 | Col 2 | Col 3 | Col 4 |
|-------|---------|---------|---------|---------|
| Row 1 | x[0][0] | x[0][1] | x[0][2] | x[0][3] |
| Row 2 | x[1][0] | x[1][1] | x[1][2] | x[1][3] |
| Row 3 | x[2][0] | x[2][1] | x[2][2] | x[2][3] |

This video is sole property of Talent Battle Pvt. Ltd. Strict penal action will be taken against unauthorized piracy of this video

C++ Arrays

Multidimensional Array Initialization

Like a normal array, we can initialize a multidimensional array in more than one way.

Initialization of two-dimensional array

```
int test[2][3] = {2, 4, 5, 9, 0, 19};
```

The above method is not preferred. A better way to initialize this array with the same array elements is given below:

| | Col 1 | Col 2 | Col 3 |
|-------|-------|-------|-------|
| Row 1 | 2 | 4 | 5 |
| Row 2 | 9 | 0 | 19 |

```
int test[2][3] = {{2, 4, 5}, {9, 0, 19}};
```

This array has 2 rows and 3 columns, which is why we have two rows of elements with 3 elements each.

This video is sole property of Talent Battle Pvt. Ltd. Strict penal action will be taken against unauthorized piracy of this video

C++ Arrays

Passing Array to a Function in C++ Programming

In C++, we can pass arrays as an argument to a function. And, also we can return arrays from a function.

Syntax for Passing Arrays as Function Parameters

The syntax for passing an array to a function is:

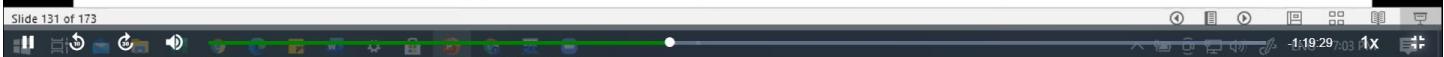
```
returnType functionName(dataType arrayName[arraySize]) {  
    // code  
}
```

Let's see an example,

```
int total(int marks[5]) {  
    // code  
}
```

Here, we have passed an int type array named marks to the function total(). The size of the array is 5.

This video is sole property of Talent Battle Pvt. Ltd. Strict penal action will be taken against unauthorized piracy of this video



```
=====  
// C++ Program to display marks of 5 students  
  
#include <iostream>  
using namespace std;  
  
// declare function to display marks  
// take a 1d array as parameter  
void display(int m[5]){  
    cout << "Displaying marks: " << endl;  
  
    // display array elements  
    for(int i = 0; i < 5; ++i){  
        cout << "Student " << i + 1 << ":" << m[i] << endl;  
    }  
}  
  
int main(){  
    // declare and initialize an array  
    int marks[5] = {88, 76, 90, 61, 69};  
  
    // call display function  
    // pass array as argument  
    display(marks);  
  
    return 0;  
}
```

```
/*
Displaying marks:
Student 1: 88
Student 2: 76
Student 3: 90
Student 4: 61
Student 5: 69

-----
Process exited after 0.06605 seconds with return value 0
*/
```

PowerPoint Slide Show - [C++ Technical Training CS] - PowerPoint

C++ Strings

String is a collection of characters.

There are two types of strings commonly used in C++ programming language:

- Strings that are objects of string class (The Standard C++ Library string class)
- C-strings (C-style Strings)

This video is sole property of Talent Battle Pvt. Ltd. Strict penal action will be taken against unauthorized piracy of this video

Slide 133 of 173

C-strings

In C programming, the collection of characters is stored in the form of arrays. This is also supported in C++ programming. Hence it's called C-strings.

C-strings are arrays of type char terminated with null character, that is, \0 (ASCII value of null character is 0).

How to define a C-string?
`char str[] = "C++";`

In the above code, str is a string and it holds 4 characters.

Although, "C++" has 3 character, the null character \0 is added to the end of the string automatically.

This video is sole property of Talent Battle Pvt. Ltd. Strict penal action will be taken against unauthorized piracy of this video

Slide 134 of 173

```
//=====
// C++ Program to display a string entered by user.
/*
The extraction operator >> works as scanf() in c and considers
a space " " has a terminating character.
*/
#include<iostream>
using namespace std;

int main(){
    char str[100];

    cout << "Enter a string: ";
    cin >> str;
    cout << "You entered: " << str << endl;

    cout << "\nEnter another string: ";
    cin >> str;
    cout << "You entered: " << str << endl;

    return 0;
}

/*
Enter a string: best
You entered: best

Enter another string: C++_language
You entered: C++_language

-----
Process exited after 16.61 seconds with return value 0
*/
```

```
//=====
// C++ program to read and display an entre line entered by user.

#include<iostream>
using namespace std;

int main(){
    char str[100];
    cout << "Enter a string: ";
    cin.get(str, 100);

    cout << "You entered: " << str << endl;
    return 0;
}

/*
Enter a string: Best language is C++ for window application
You entered: Best language is C++ for window application

-----
Process exited after 63.44 seconds with return value 0
Press any key to continue . . .
*/
```

PowerPoint Slide Show - [C++ Technical Training CS] - PowerPoint

C++ Strings

To read the text containing blank space, `cin.get` function can be used. This function takes two arguments.

First argument is the name of the string (address of first element of string) and second argument is the maximum size of the array.

In the above program, str is the name of the string and 100 is the maximum size of the array.

This video is sole property of Talent Battle Pvt. Ltd. Strict penal action will be taken against unauthorized piracy of this video

Slide 137 of 173

PowerPoint Slide Show - [C++ Technical Training CS] - PowerPoint

C++ Strings

string Object

In C++, you can also create a string object for holding strings.

Unlike using char arrays, string objects has no fixed length, and can be extended as per your requirement.

This video is sole property of Talent Battle Pvt. Ltd. Strict penal action will be taken against unauthorized piracy of this video

Slide 138 of 173

PowerPoint Slide Show - [C++ Technical Training CS] - PowerPoint

```
//=====
/*
In this program, a string str is declared. Then the
string is asked from the user

Instead of using cin >> or cin.get() function,
you can get the entered line of text using getline().

getline() function takes the input stream as the first
parameter which is cin and str as the location of the line
to be stored.

*/
#include<iostream>
using namespace std;

int main(){
    // declaring a string object
    string str;
    cout << "Enter a string: ";
    getline(cin, str);

    cout << "You entered: " << str << endl;
    return 0;
}

/*
Enter a string: This is the best programming language. I want to crack Google as a software
engineer.
You entered: This is the best programming language. I want to crack Google as a software
engineer.

-----
Process exited after 41.87 seconds with return value 0
*/
```

C++ Program to Calculate Average of Numbers Using Arrays
C++ Program to Find Largest Element of an Array
C++ Program to Calculate Standard Deviation
C++ Program to Add Two Matrix Using Multi-dimensional Arrays
C++ Program to Multiply Two Matrix Using Multi-dimensional Arrays
C++ Program to Find Transpose of a Matrix
C++ Program to Multiply two Matrices by Passing Matrix to Function
C++ Program to Access Elements of an Array Using Pointer
C++ Program to Swap Numbers in Cyclic Order Using Call by Reference
C++ Program to Find the Frequency of Characters in a String
C++ Program to Find the Number of Vowels, Consonants, Digits and White Spaces in a String
C++ Program to Remove all Characters in a String Except Alphabets.
C++ Program to Find the Length of a String
C++ Program to Concatenate Two Strings
C++ Program to Copy Strings
C++ Program to Sort Elements in Lexicographical Order (Dictionary Order)

This video is sole property of Talent Battle Pvt. Ltd. Strict penal action will be taken against unauthorized piracy of this video

PowerPoint Slide Show - [C++ Technical Training CS] - PowerPoint

C++ Structures

Structure is a collection of variables of different data types under a single name. It is similar to a **class** in that, both holds a collection of data of different data types.

How to declare a structure in C++ programming?

The struct keyword defines a structure type followed by an identifier (name of the structure). Then inside the curly braces, you can declare one or more members (declare variables inside curly braces) of that structure. For example:

```
struct Person
{
    char name[50];
    int age;
    float salary;
};
```

Here a structure person is defined which has three members: name, age and salary.

When a structure is created, no memory is allocated.

This video is sole property of Talent Battle Pvt. Ltd. Strict penal action will be taken against unauthorized piracy of this video

Slide 141 of 173

PowerPoint Slide Show - [C++ Technical Training CS] - PowerPoint

C++ Structures

How to define a structure variable?

Once you declare a structure person as above. You can define a structure variable as:

```
Person bill;
```

Here, a structure variable bill is defined which is of type structure Person.

When structure variable is defined, only then the required memory is allocated by the compiler.

Considering you have either 32-bit or 64-bit system, the memory of float is 4 bytes, memory of int is 4 bytes and memory of char is 1 byte.

Hence, 58 bytes of memory is allocated for structure variable bill.

This video is sole property of Talent Battle Pvt. Ltd. Strict penal action will be taken against unauthorized piracy of this video

C++ Structures

How to access members of a structure?

The members of structure variable is accessed using a dot (.) operator.

Suppose, you want to access age of structure variable bill and assign it 50 to it. You can perform this task by using following code below:

```
bill.age = 50;
```

This video is sole property of Talent Battle Pvt. Ltd. Strict penal action will be taken against unauthorized piracy of this video

```
//=====
// C++ structures

#include<iostream>
using namespace std;

struct Person {
    char name[50];
    int age;
    float salary;
};

int main(){
    Person p1;

    cout << "Enter Full name: ";
    cin.get(p1.name, 50);

    cout << "Enter age: ";
    cin >> p1.age;

    cout << "Enter salary: ";
    cin >> p1.salary;

    cout << "\nDisplaying Information."<<endl;
    cout << "Name: " << p1.name << endl;
    cout << "Age: " << p1.age << endl;
    cout << "Salary: " << p1.salary;

    return 0;
}

/*
Enter Full name: Dheeraj Kumar
Enter age: 22
Enter salary: 900000

Displaying Information.
Name: Dheeraj Kumar
Age: 22
Salary: 900000
-----
Process exited after 24.99 seconds with return value 0
*/
```

C++ Structures

C++ Structure and Function

Passing structure to function in C++

A structure variable can be passed to a function in similar way as normal argument. Consider this example:

```
#include <iostream>
using namespace std;
struct Person {
    char name[50];
    int age;
    float salary;
};
void displayData(Person); // Function declaration
int main() {
    Person p;
    cout << "Enter Full name: ";
    cin.get(p.name, 50);
    cout << "Enter age: ";
    cin >> p.age;
    cout << "Enter salary: ";
    cin >> p.salary;
    // Function call with structure variable as an argument
    displayData(p);
    return 0;
}
void displayData(Person p) {
    cout << "\nDisplaying Information." << endl;
    cout << "Name: " << p.name << endl;
    cout << "Age: " << p.age << endl;
    cout << "Salary: " << p.salary;
}
```

This video is sole property of Talent Battle Pvt. Ltd. Strict penal action will be taken against unauthorized piracy of this video

```
//=====
// C++ Structure and function
// passing structure to function in C++
/*
A stucture variable can be passed to a function in similar
way as normal argument. Consider this example:
*/

#include<iostream>
using namespace std;

struct Person {
    char name[50];
    int age;
    float salary;
};

void displayData(Person); // Function declaration

int main(){
    Person p;
    cout << "Enter full name: ";
    cin.get(p.name, 50);
    cout << "Enter age: ";
    cin >> p.age;
    cout << "Enter salary: ";
    cin >> p.salary;

    // function call with structure variable as an argument
    displayData(p);
    return 0;
}

void displayData(Person p){
    cout << "\nDisplaying Information." << endl;
    cout << "Name: " << p.name << endl;
    cout << "Age: " << p.age << endl;
    cout << "Salary: " << p.salary;
}

/*
Enter full name: Dheeraj
Enter age: 22
Enter salary: 900000.99

Displaying Information.
Name: Dheeraj
Age: 22
Salary: 900001
-----
```

```
Process exited after 22.84 seconds with return value 0
```

```
*/
```

C++ Structures

C++ Pointers to Structure

A pointer variable can be created not only for native types like (int, float, double etc.) but they can also be created for user defined types like structure.

Here is how you can create pointer for structures:

```
#include <iostream>
using namespace std;
struct temp {
    int i;
    float f;
};
int main() {
    temp *ptr;
    return 0;
}
```

This video is sole property of Talent Battle Pvt. Ltd. Strict penal action will be taken against unauthorized piracy of this video

```
=====
```

```
// C++ Pointers to structure
/*
A pointer variable can be created not only for native types like
(int, float, double etc.) but they can also be created for user
defined types like structure.
*/
```

```
// Here is how you can create pointer for structures:
```

```
#include<iostream>
using namespace std;
struct temp {
    int i;
    float f;
};

int main(){
    temp *ptr;
    return 0;
}
```

```
//=====
// C++ structure

#include<iostream>
using namespace std;
struct Distance{
    int feet;
    float inch;
};

int main(){
    Distance *ptr, d;
    ptr = &d;
    cout << "Enter feet: ";
    cin >> (*ptr).feet;
    cout << "Enter inch: ";
    cin >> (*ptr).inch;

    cout << "Displaying information." << endl;
    cout << "Distance = " << (*ptr).feet << " feet " << (*ptr).inch << "inches";

    return 0;
}

/*
Enter feet: 5.5
Enter inch: Displaying information.
Distance = 5 feet 0.5inches
-----
Process exited after 2.339 seconds with return value 0
*/
```

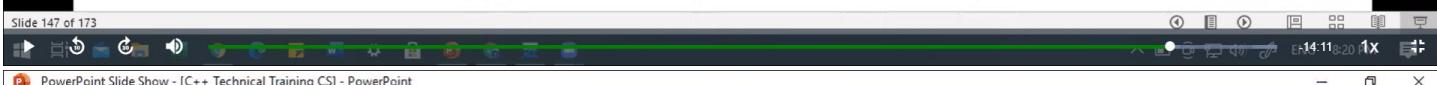
C++ Structures

```
#include <iostream>
using namespace std;
struct Distance {
    int feet;
    float inch;
};
int main() {
    Distance *ptr, d;
    ptr = &d;
    cout << "Enter feet: ";
    cin >> (*ptr).feet;
    cout << "Enter inch: ";
    cin >> (*ptr).inch;
    cout << "Displaying information." << endl;
    cout << "Distance = " << (*ptr).feet << " feet " <<
        (*ptr).inch << " inches";
    return 0;
}
```

Annotations on the code:

- Red arrows point from the variable names `feet` and `inch` to the text "pointer variable".
- Red arrows point from the variable names `ptr` and `d` to the text "normal or.. variable".
- Red annotations show the equivalence:
 - `d.feet` is equivalent to `(*ptr).feet`
 - `d.inch` is approximately equal to `(*ptr).inch`

This video is sole property of Talent Battle Pvt. Ltd. Strict penal action will be taken against unauthorized piracy of this video



C++ Structures

Notes:

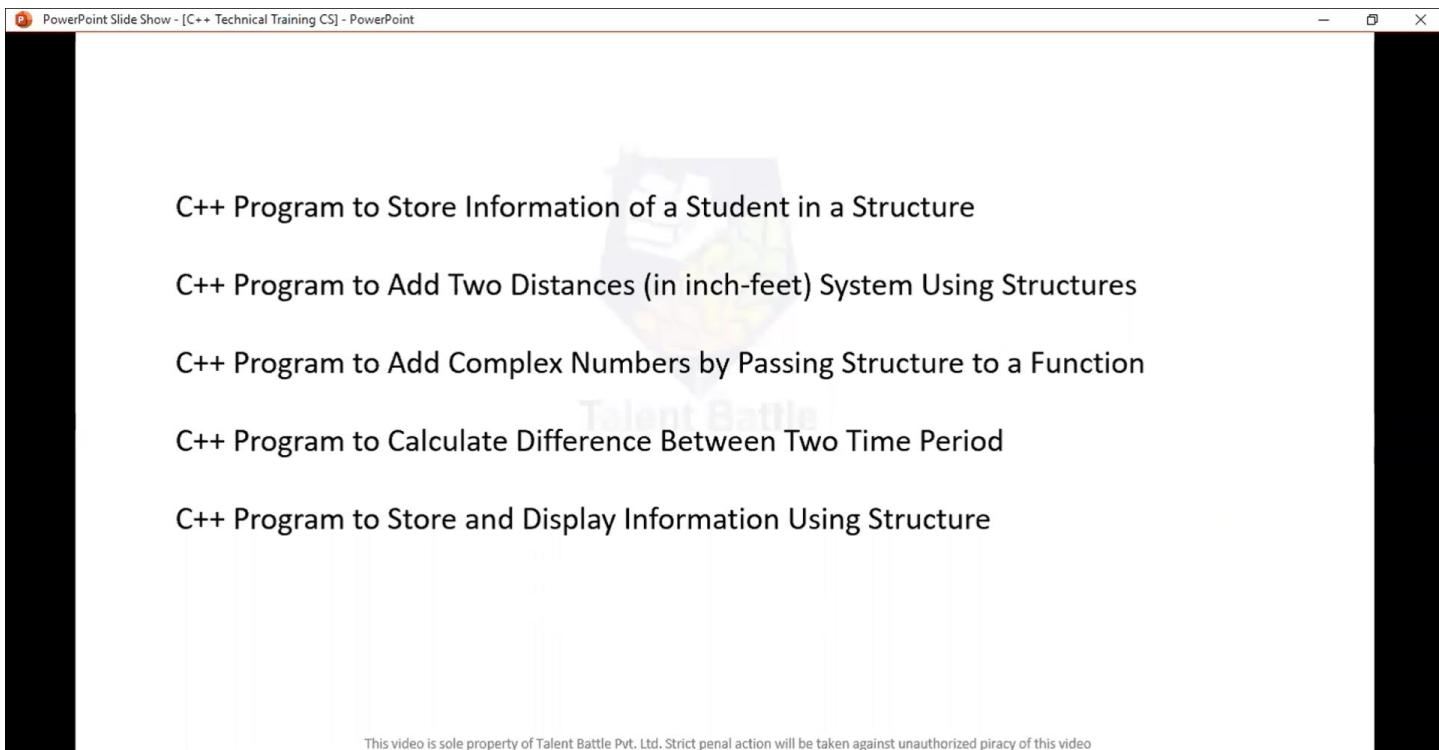
Since pointer `ptr` is pointing to variable `d` in this program, `(*ptr).inch` and `d.inch` are equivalent. Similarly, `(*ptr).feet` and `d.feet` are equivalent.

However, if we are using pointers, it is far more preferable to access class members using the `->` operator. This is because the `.` operator has a higher precedence than the `*` operator. Hence, we enclose `*ptr` in brackets when using `(*ptr).inch`. Because of this, it is easier to make mistakes if both operators are used together in a single code.

`ptr->feet` is same as `(*ptr).feet`
`ptr->inch` is same as `(*ptr).inch`

This video is sole property of Talent Battle Pvt. Ltd. Strict penal action will be taken against unauthorized piracy of this video





This video is sole property of Talent Battle Pvt. Ltd. Strict penal action will be taken against unauthorized piracy of this video

C++ Enumeration

An enumeration is a user-defined data type that consists of integral constants. To define an enumeration, keyword enum is used.

enum season { spring, summer, autumn, winter };

Here, the name of the enumeration is season.

And, spring, summer and winter are values of type season.

By default, spring is 0, summer is 1 and so on. You can change the default value of an enum element during declaration (if necessary).

This video is sole property of Talent Battle Pvt. Ltd. Strict penal action will be taken against unauthorized piracy of this video

