CodeWithHarry

🏠   HTML   CSS   JS   C   C++   JAVA   PYTHON   PHP   🔍

Array of Objects & Passing Objects as Function Arguments in C++ | C++ Tutorials for Beginners #25

▶

Overview   Q&A   Downloads   Announcements

Array of Objects & Passing Objects as Function Arguments in C++ | C++ Tutorials for Beginners #25

In this tutorial, we will discuss an array of objects and passing objects as a function arguments in C++

An array of Objects in C++

An array of objects is declared the same as any other data-type array. An array of objects consists of class objects as its elements. If the array consists of class objects it is called an array of objects. An example program to demonstrate the concept of an array of objects is shown below.

```
class Employee
```

```cpp
{
    int id;
    int salary;

public:
    void setId(void)
    {
        salary = 122;
        cout << "Enter the id of employee" << endl;
        cin >> id;
    }

    void getId(void)
    {
        cout << "The id of this employee is " << id << endl;
    }
};
```

Code Snippet 1: Employee Class

As shown in Code Snippet 1, we created an employee class that has integer "id" variable and "salary" integer variable as private class members; and "setId" void function, "getId" void function as public class members. These functions are explained below.

We have defined a "setId" function. In this function, the "salary" variable is assigned by the value "122" and the function will take input for "id" from the user at runtime. We have defined a "getId" function. This function will print the values of the variables "id".
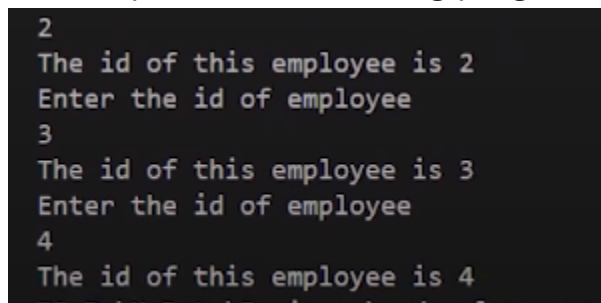
```cpp
int main()
{
    Employee fb[4];
    for (int i = 0; i < 4; i++)
    {
        fb[i].setId();
        fb[i].getId();
    }

    return 0;
}
```

Code Snippet 2: main program

As shown in Code Snippet 2, we created an array "fb" of size "4" which is of employee data-type. The "for" loop is used to run "setId" and "getId" functions till the size of an array. The main thing to note here is that the objects can also be created individually but it is more convenient to use an array if too many objects are to be created. The output of the following program is shown in figure 1.

```
2
The id of this employee is 2
Enter the id of employee
3
The id of this employee is 3
Enter the id of employee
4
The id of this employee is 4
```

Figure 1: Employee Program Output

As shown in figure 1. As we input the Id for an employee it gives us the output of the employee Id.

Passing Object as Function Argument

Objects can be passed as function arguments. This is useful when we want to assign the values of a passed object to the current object. An example program to demonstrate the concept of passing an object as a function argument is shown below.

```cpp
class complex{
    int a;
    int b;

    public:
        void setData(int v1, int v2){
            a = v1;
            b = v2;
        }

        void setDataBySum(complex o1, complex o2){
            a = o1.a + o2.a;
            b = o1.b + o2.b;
        }

        void printNumber(){
            cout<<"Your complex number is "<<a<<" + "<<b<<"i"<<endl;
        }
};
```

Code Snippet 3: Complex Class

As shown in Code Snippet 3, we created a complex class that has integer "a" variable and "b" integer variable as private class members; and "setData" void function, "setDataBySum" void function, and "printNumber" void function as public class members. These functions are explained below.

We have defined a "setData" function. In this function the values are assigned to the variables "a" and "b" because they are private data members of the class and values cannot be assigned directly. We have defined a "setDataBySum" function. In this function, the values of two objects are added and then assigned to the variables "a" and "b". We have defined a "printNumber" function. In this function, the values of the variable "a" and "b" are being printed.

```cpp
int main(){
    complex c1, c2, c3;
    c1.setData(1, 2);
    c1.printNumber();

    c2.setData(3, 4);
    c2.printNumber();

    c3.setDataBySum(c1, c2);
    c3.printNumber();
    return 0;
}
```
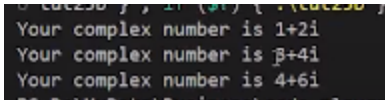
Code Snippet 4: main program 2

As shown in Code Snippet 4:

- We have created object "c1", "c2", and"c3" of complex data-type.
- The object "c1" calls the "setData" and "printNumber" functions.

- The object "c2" calls the "setData" and "printNumber" functions.
- The object "c3" calls the "setDataBySum" and "printNumber" functions.

The output of the following program is shown in figure 2.

Figure 2: Complex Program Output

Code as described/written in the video

```cpp
#include <iostream>
using namespace std;

class Employee
{
    int id;
    int salary;


public:
    void setId(void)
    {
        salary = 122;
        cout << "Enter the id of employee" << endl;
        cin >> id;
    }

    void getId(void)
    {
```

```cpp
            cout << "The id of this employee is " << id << endl;
        }
    };

    int main()
    {
        // Employee harry, rohan, lovish, shruti;
        // harry.setId();
        // harry.getId();
        Employee fb[4];
        for (int i = 0; i < 4; i++)
        {
            fb[i].setId();
            fb[i].getId();
        }

        return 0;
    }
```

Code 25b as described/written in the video

```cpp
#include<iostream>
using namespace std;

class complex{
    int a;
```

```cpp
        int b;

    public:
        void setData(int v1, int v2){
            a = v1;
            b = v2;
        }


        void setDataBySum(complex o1, complex o2){
            a = o1.a + o2.a;
            b = o1.b + o2.b;
        }


        void printNumber(){
            cout<<"Your complex number is "<<a<<" + "<<b<<"i"<<endl;
        }
};

int main(){
    complex c1, c2, c3;
    c1.setData(1, 2);
    c1.printNumber();

    c2.setData(3, 4);
    c2.printNumber();
```

```cpp
    c3.setDataBySum(c1, c2);
    c3.printNumber();
    return 0;
}
```

Previous

Next

CodeWithHarry   Copyright © 2022 CodeWithHarry.com