



Array of Objects Using Pointers in C++ | C++ Tutorials for Beginners #52



[Overview](#) [Q&A](#) [Downloads](#) [Announcements](#)

Array of Objects Using Pointers in C++ | C++ Tutorials for Beginners #52

In this tutorial, we will discuss an array of objects using pointers in C++

Array of Objects Using Pointers in C++

Array of objects can be defined as an array that's each element is an object of the class. In this tutorial, we will use the pointer to store the address of an array of objects. An example program is shown below to demonstrate the concept of an array of objects using pointers.

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

```
class ShopItem
{
    int id;
    float price;
public:
    void setData(int a, float b){
        id = a;
        price = b;
    }
    void getData(void){
        cout<<"Code of this item is "<< id<<endl;
        cout<<"Price of this item is "<<price<<endl;
    }
};
```

Code Snippet 1: Array of Objects Using Pointers Example Program

As shown in a code snippet 1,

1. We created a class “ShopItem”, which contains two private data members “id” and “price”.
2. The class “ShopItem” contains two member functions “setdata” and “getdata”
3. The Function “setdata” will take two parameters and assign the values of parameters to the private data members “id” and “price”
4. The Function “getdata” will print the values of private data members “id” and “price”

```
int main(){
    int size = 3;
```

```
ShopItem *ptr = new ShopItem [size];
ShopItem *ptrTemp = ptr;
int p, i;
float q;
for (i = 0; i < size; i++)
{
    cout<<"Enter Id and price of item "<< i+1<<endl;
    cin>>p>>q;
    // (*ptr).setData(p, q);
    ptr->setData(p, q);
    ptr++;
}

for (i = 0; i < size; i++)
{
    cout<<"Item number: "<<i+1<<endl;
    ptrTemp->getData();
    ptrTemp++;
}

return 0;
}
```

Code Snippet 2: Main Program

As shown in code snippet 2,

1. We created an integer variable “size” and assigned the value “3” to it.
2. Array of objects of size “3” is created dynamically by using the “new” keyword and its address is assigned to the pointer “ptr”
3. The address of pointer “ptr” is assigned to another pointer “ptrTemp”
4. Two integer variables “p” and “i” are declared and one float variable “q” is declared
5. We created a “for” loop which will run till the size of array and will take input for “id” and “price” from user at run time. In this “for” loop “setdata” function is called using pointer “ptr”; the function will set the values of “id” and “price” which user will enter. The value of the pointer “ptr” is incremented by 1 in every iteration of loop.
6. We created another “for” loop which will run till the size of array and will print the number of the item. In this “for” loop “getdata” function is called using pointer “ptr”; the function will print the values of “id” and “price”. The value of the pointer “ptrTemp” is incremented by 1 in every iteration of loop.

The main thing to note here is that in the first “for” loop we are incrementing the value of the pointer “ptr” because it is pointing to the address of array of objects and when loop will run every time the function “setdata” will be called by the different object. If we don’t increment the value of the pointer “ptr” the each time function “setdata” will be called by the same object. Likewise in the second loop we are incrementing the pointer “ptrTemp” so that the function “getdata” could be called by each object in the array.

The input and output of the following program is shown below,

```
Enter Id and price of item 1
1001
1.1
Enter Id and price of item 2
1002
1.2
Enter Id and price of item 3
1003
3.3
```

Figure 1: Array of Objects Using Pointer Program Input

```
Item number: 1  
Code of this item is 1001  
Price of this item is 1.1  
Item number: 2  
Code of this item is 1002  
Price of this item is 1.2  
Item number: 3  
Code of this item is 1003  
Price of this item is 3.3
```

Figure 2: Array of Objects Using Pointer Program Output

[Previous](#)[Next](#)

CodeWithHarry

Copyright © 2022 CodeWithHarry.com

