C++ POINTER TO AN ARRAY

It is most likely that you would not understand this chapter until you go through the chapter related C++ Pointers.

So assuming you have bit understanding on pointers in C++, let us start: An array name is a constant pointer to the first element of the array. Therefore, in the declaration:

```
double balance[50];
```

balance is a pointer to &balance[0], which is the address of the first element of the array balance. Thus, the following program fragment assigns **p** the address of the first element of **balance**:

```
double *p;
double balance[10];

p = balance;
```

It is legal to use array names as constant pointers, and vice versa. Therefore, $*balance + 4$ is a legitimate way of accessing the data at balance[4].

Once you store the address of first element in p, you can access array elements using *p, $*p + 1$, $*p + 2$ and so on. Below is the example to show all the concepts discussed above:

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

int main ()
{
   // an array with 5 elements.
   double balance[5] = {1000.0, 2.0, 3.4, 17.0, 50.0};
   double *p;

   p = balance;

   // output each array element's value
   cout << "Array values using pointer " << endl;
   for ( int i = 0; i < 5; i++ )
   {
       cout << "*(p + " << i << ") : ";
       cout << *(p + i) << endl;
   }

   cout << "Array values using balance as address " << endl;
   for ( int i = 0; i < 5; i++ )
   {
       cout << "*(balance + " << i << ") : ";
       cout << *(balance + i) << endl;
   }

   return 0;
}
```

When the above code is compiled and executed, it produces the following result:

```
Array values using pointer
*(p + 0) : 1000
*(p + 1) : 2
*(p + 2) : 3.4
*(p + 3) : 17
*(p + 4) : 50
Array values using balance as address
*(balance + 0) : 1000
```

```
*(balance + 1) : 2
*(balance + 2) : 3.4
*(balance + 3) : 17
*(balance + 4) : 50
```

In the above example, p is a pointer to double which means it can store address of a variable of double type. Once we have address in p, then ***p** will give us value available at the address stored in p, as we have shown in the above example.

Loading [MathJax]/jax/output/HTML-CSS/fonts/TeX/fontdata.js