COMPUTER SCIENCE : C++

Passing Array to Function

You can pass array as an argument to a function just like you pass variables as arguments. In order to pass array to the function you just need to **mention the array name during function call** like this:

```
function_name(array_name);
```

Passing arrays to a function

In this example, we are passing two arrays a & b to the function sum(). This function adds the corresponding elements of both the arrays and display them.

```
#include <iostream>
using namespace std;
/* This function adds the corresponding
 * elements of both the arrays and
 * displays it.
 */
void sum(int arr1[], int arr2[]){
   int temp[5];
   for(int i=0; i<5; i++){
      temp[i] = arr1[i]+arr2[i];
```

```
      cout<<temp[i]<<endl;
   }
}
int main(){
   int a[5] = {10, 20, 30, 40 ,50};
   int b[5] = {1, 2, 3, 4, 5};
   //Passing arrays to function
   sum(a, b);
   return 0;
}
```

Output:

```
11
22
33
44
55
```

Passing multidimensional array to function

In this example we are passing a multidimensional array to the function square which displays the square of each element.

```
#include <iostream>
#include <cmath>
using namespace std;
/* This method prints the square of each
 * of the elements of multidimensional array
 */
void square(int arr[2][3]){
   int temp;
```

```
   for(int i=0; i<2; i++){
      for(int j=0; j<3; j++){
        temp = arr[i][j];
        cout<<pow(temp, 2)<<endl;
      }
   }
}
int main(){
   int arr[2][3] = {
       {1, 2, 3},
       {4, 5, 6}
   };
   square(arr);
   return 0;
}
```

Output:

```
1
4
9
16
25
36
```

Strings in C++

Strings are words that are made up of characters, hence they are known as sequence of characters. In C++ we have two ways to create and use strings: 1) By creating char arrays and treat them as string 2) By creating `string` object

Lets discuss these two ways of creating string first and then we will see which method is better and why.

1) Array of Characters - Also known as C Strings

A simple example where we have initialized the char array during declaration.

```
#include <iostream>
using namespace std;
int main(){
   char book[50] = "A Song of Ice and Fire";
   cout<<book;
   return 0;
}
```

Output:

```
A Song of Ice and Fire
```

: Getting user input as string

This can be considered as inefficient method of reading user

input, why? Because when we read the user input string using `cin` then only the first word of the string is stored in char array and rest get ignored. The cin function considers the space in the string as delimiter and ignore the part after it.

```
#include <iostream>
using namespace std;
int main(){
   char book[50];
   cout<<"Enter your favorite book name:";
   //reading user input
   cin>>book;
   cout<<"You entered: "<<book;
   return 0;
}
```

Output:

```
Enter your favorite book name:The Murder of Roger Ackroyd
You entered: The
```

You can see that only the “The” got captured in the book and remaining part after space got ignored. How to deal with this then? Well, for this we can use `cin.get` function, which reads the complete line entered by user.

Correct way of capturing user input string using cin.get

```
#include <iostream>
using namespace std;
int main(){
   char book[50];
   cout<<"Enter your favorite book name:";

   //reading user input
   cin.get(book, 50);
   cout<<"You entered: "<<book;
   return 0;
}
```

Output:

```
Enter your favorite book name:The Murder of Roger
Ackroyd
You entered: The Murder of Roger Ackroyd
```

Drawback of this method

1) Size of the char array is fixed, which means the size of the string created through it is fixed in size, more memory cannot be allocated to it during runtime. For example, lets say you have created an array of character with the size 10 and user enters the string of size 15 then the last five characters would be truncated from the string.
On the other hand if you create a larger array to accommodate user input then the memory is wasted if the user input is small and array is much larger then what is needed.

2) In this method, you can only use the in-built functions created for array which don't help much in string manipulation.

String object in C++

Till now we have seen how to handle strings in C++ using char arrays. Lets see another and better way of handling strings in C++ – string objects.

```
#include<iostream>
```

```
using namespace std;
int main(){
   // This is how we create string object
   string str;
   cout<<"Enter a String:";
   /* This is used to get the user input
    * and store it into str
    */
   getline(cin,str);
   cout<<"You entered: ";
   cout<<str<<endl;

   /* This function adds a character at
    * the end of the string
    */ str.push_back('A');
   cout<<"The string after push_back: "<<str<<endl;
   /* This function deletes a character from
    * the end of the string
    */
   str.pop_back();
   cout << "The string after pop_back: "<<str<<endl;
   return 0;
}
```

Output:

```
Enter a String:XYZ
```

```
You entered: XYZ
The string after push_back: XYZA
The string after pop_back: XYZ
```

The advantage of using this method is that you need not to declare the size of the string, the size is determined at run time, so this is better memory management method. The memory is allocated dynamically at runtime so no memory is wasted.