**Passing Array to a Function in C++ Programming**

**Riya Widayanti**

Setelah mempelajari array dan fungsi dalam minggu sebelumnya, saat ini akan dipelajari bagaimana dalam fungsi memiliki passig parameter yang sebelumnya variabel tunggal, namun saat ini menggunakan variabel array. Inget passing parameternya menggunakan variabel Array



Deklarasi variabel untu fungsi yang memiliki passing parameter variabel tunggal

Tipe\_data nama\_variabel\_fungsi(tipe\_data nama\_variabel passing);

Contoh

Int hitung\_luas(int panjang, int lebar);

Artinya

Bahwa fungsi hitung luas memiliki passing parameter 2 yaitu variabel tunggal bertipe integer dengan nama variabel panjang dan lebar

#include <iostream>

using namespace std;

int hitung\_luas(int p, int l);

int main()

{

 int panjang, lebar, luas;

 cin>>panjang;

 cin>>lebar;

 luas = hitung\_luas(panjang, lebar);

 cout<<luas;

 return 0;

}

int hitung\_luas(int p, int l)

{

 return p\*l;

}

Sedangkan untuk passing parameter dengan variabel array, deklarasi variabelnya adalah

Tipe\_data nama\_varibel\_fungsi(tipe\_data nama\_variebel\_array)

Contoh:**Example 1: Passing One-dimensional Array to a Function**

**P**rogram di bawah ini bertujuan untuk mencetak varibel array

#include <iostream>

using namespace std;

void display(int marks[5]);

int main()

{

 int marks[5] = {88, 76, 90, 61, 69};

 display(marks);

 return 0;

}

void display(int m[5])

{

 cout << "Displaying marks: "<< endl;

 for (int i = 0; i < 5; ++i)

 {

 cout << "Student "<< i + 1 <<": "<< m[i] << endl;

 }

}

**Output**

Displaying marks:

Student 1: 88

Student 2: 76

Student 3: 90

Student 4: 61

Student 5: 69

Dalam pemahaman di atas untuk memanggil fungsi dengan passing varibel array cukup dengan nama\_fungsi yang diikuti nama\_variebel\_Array nya

display(marks);

Perhatikan dalam deklarasi fungsi dengan passing parameter varibel tunggul dengan varibel array yang membedakan tentunya ada indeksnya saja.

void display(int m[5]);

The argument marks in the above code represents the memory address of first element of array marks[5].

And the formal argument int m[5] in function declaration converts to int\* m;. This pointer points to the same address pointed by the array marks.

That's the reason, although the function is manipulated in the user-defined function with different array name m[5], the original array marks is manipulated.

C++ handles passing an array to a function in this way to save memory and time.

**Passing Multidimensional Array to a Function**

[Multidimensional array](https://www.programiz.com/cpp-programming/multidimensional-arrays) can be passed in similar way as one-dimensional array. Consider this example to pass two dimensional array to a function:

**Example 2: Passing Multidimensional Array to a Function**

**C++ Program to display the elements of two dimensional array by passing it to a function.**

#include <iostream>

using namespace std;

void display(int n[3][2]);

int main()

{

 int num[3][2] = {

 {3, 4},

 {9, 5},

 {7, 1}

 };

 display(num);

 return 0;

}

void display(int n[3][2])

{

 cout << "Displaying Values: " << endl;

 for(int i = 0; i < 3; ++i)

 {

 for(int j = 0; j < 2; ++j)

 {

 cout << n[i][j] << " ";

 }

 }

}

**Output**

Displaying Values:

3 4 9 5 7 1

In the above program, the multi-dimensional array num is passed to the function display().

Inside, display() function, the array n (num) is traversed using a nested for loop.

The program uses 2 [for loops](https://www.programiz.com/cpp-programming/for-loop) to iterate over the elements inside a 2-dimensional array. If it were a 3-dimensional array, you should use 3 for loops.

Finally, all elements are printed onto the screen.

**Note:**Multidimensional array with dimension more than 2 can be passed in similar way as two dimensional array.

Berikutnya akan dipelajari tentang String.



String merupakan kumpulan dari karakter. Ada 2 tipe dalam string yang sering digunakan dalam C++ programming language:

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

**C-strings**

Daalm bahasa C kumpulan karakter disimpan dalam bentuk Array, hal ini yang disebut dengan C-String. Dimana nilai akhir dari array yang berisi kareakter diakhiri dengan nilai null karakter, hal ini merupakan kodse ASCI dari \0.

Cara Mendefinisikan varibel c-string

char str[] = "C++";

Contoh di atas adalah varibel str adalah variabel string dan memiliki 4 karakter, meskipun secara “C++” memiliki 3 karakter, namun karakter terakhir selalu diisi \0 secara default.

**Definisi lain String**

char str[4] = "C++";

char str[] = {'C','+','+','\0'};

char str[4] = {'C','+','+','\0'};

Like arrays, it is not necessary to use all the space allocated for the string. For example:

char str[100] = "C++";

**Example 1: C++ String to read a word**

**C++ program to display a string entered by user.**

#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;

}

**Output**

Enter a string: C++

You entered: C++

Enter another string: Programming is fun.

You entered: Programming

Dalam contoh program di atas, merupakan contoh inputan string dan mencetaknya langsung. Ada dua string ang diinput yaitu C++ dan hasilnya juga c++. Namun ketika diinput "Programming is fun". Yang tampil mengapa Programming saja, karena dalam string yang kedua mengandung spasi oleh c++ dianggap sebagai enter atau terminatiing character. (akhir dari karakter)

**Example 2: C++ String to read a line of text**

#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;

}

**Output**

Enter a string: Programming is fun.

You entered: Programming is fun.

Pada contoh kali ini text yang mengandung spasi, kita input dengan menggunakan fungsi cin.get, yang memiliki 2 argumen yaitu:

Argumen pertama adalah nama variabel stringnya dan kedua adalah ukuran maksimun dari varibel string. Untuk contoh di atas berarti str adalah nama varibel sedangkan maksimum memnampung karakter 100 (termasuk null karakter ya)

**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.

**Example 3: C++ string using string data type**

#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;

}

**Output**

Enter a string: Programming is fun.

You entered: Programming is fun.

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.

**Passing String to a Function**

Strings are passed to a function in a similar way [arrays are passed to a function](https://www.programiz.com/cpp-programming/passing-arrays-function).

#include <iostream>

using namespace std;

void display(char \*);

void display(string);

int main()

{

 string str1;

 char str[100];

 cout << "Enter a string: ";

 getline(cin, str1);

 cout << "Enter another string: ";

 cin.get(str, 100, '\n');

 display(str1);

 display(str);

 return 0;

}

void display(char s[])

{

 cout << "Entered char array is: " << s << endl;

}

void display(string s)

{

 cout << "Entered string is: " << s << endl;

}

**Output**

Enter a string: Programming is fun.

Enter another string: Really?

Entered string is: Programming is fun.

Entered char array is: Really?

In the above program, two strings are asked to enter. These are stored in str and str1respectively, where str is a char array and str1 is a string object.

Then, we have two functions display() that outputs the string onto the string.

The only difference between the two functions is the parameter. The first display() function takes char array as a parameter, while the second takes string as a parameter.

# C++ Structures

Structure adalah kumpulan dari variabel yang memiliki tipe data yang berbeda. Ya yang perlu digaribawahi adalah “BERBEDA” yang kita sebut dengan heterogen, sedangkan dalam Array tentunya kita sebut dengan homogen. Berikut deklarasi varibel dalam structure



Istilah dalam structure kita sebut record, yang biasanya menyimpan beberapa infomrasi. Misalnya ingin menyimpan record seseorang yang terdiri dari name, citizenship number and salary. Selanjutnya Anda dapat membuat dari record tersebut yang kita sebut filed dimana masing-masing filed memiliki tipe data yang berbeda

name, citNo, salary.

Selain untuk mendefinsikan dalam satu record tertentu, pemanfaatan structure untuk sejumlah record. Yang dimaksudkan adalah

name1, citNo1, salary1, name2, citNo2, salary2

## 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.

The structure definition is only the blueprint for the creating of variables. You can imagine it as a datatype. When you define an integer as below:

int foo;

The int specifies that, variable foo can hold integer element only. Similarly, structure definition only specifies that, what property a structure variable holds when it is defined.

**Note:**Remember to end the declaration with a semicolon **(;)**

## 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.

## 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;

## Example: C++ Structure

C++ Program to assign data to members of a structure variable and display it.

#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;

}

**Output**

Enter Full name: Magdalena Dankova

Enter age: 27

Enter salary: 1024.4

Displaying Information.

Name: Magdalena Dankova

Age: 27

Salary: 1024.4

Here a structure Person is declared which has three members name, age and salary.

Inside main() [function](https://www.programiz.com/cpp-programming/function), a structure variable p1 is defined. Then, the user is asked to enter information and data entered by user is displayed.

## Passing structure to function in C++

Berikutnya akan diperlajari bila dalam fungsi terdapat varibel passing parameter yang bertipe stucture. Contohnya adalah sebagai berikut:

Example 1: C++ Structure and Function

#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;

}

**Output**

Enter Full name: Bill Jobs

Enter age: 55

Enter salary: 34233.4

Displaying Information.

Name: Bill Jobs

Age: 55

Salary: 34233.4

In this program, user is asked to enter the name, age and salary of a Person inside main()function.

Then, the structure variable p is to passed to a function using.

displayData(p);

The return type of displayData() is void and a single argument of type structure Person is passed.

Then the members of structure p is displayed from this function.

### Example 2: Returning structure from function in C++

#include <iostream>

using namespace std;

struct Person {

 char name[50];

 int age;

 float salary;

};

Person getData(Person);

void displayData(Person);

int main()

{

 Person p;

 p = getData(p);

 displayData(p);

 return 0;

}

Person getData(Person p) {

 cout << "Enter Full name: ";

 cin.get(p.name, 50);

 cout << "Enter age: ";

 cin >> p.age;

 cout << "Enter salary: ";

 cin >> p.salary;

 return p;

}

void displayData(Person p)

{

 cout << "\nDisplaying Information." << endl;

 cout << "Name: " << p.name << endl;

 cout <<"Age: " << p.age << endl;

 cout << "Salary: " << p.salary;

}

The output of this program is same as program above.

In this program, the structure variable p of type structure Person is defined under main()function.

The structure variable p is passed to getData() function which takes input from user which is then returned to main function.

p = getData(p);

**Note:** The value of all members of a structure variable can be assigned to another structure using assignment operator **=** if both structure variables are of same type. You don't need to manually assign each members.

Then the structure variable p is passed to displayData() function, which displays the information.

**Array of Structures**

Dalam array of structre ini dimaksudkan bahwa data structure digunakan bukan untuk varibel tunggal. Contohnya

#include <iostream>

#include <cstring>

using namespace std;

struct student

{

 int roll\_no;

 string name;

 int phone\_number;

};

int main(){

 student p1 = {1,"Brown",123443};

 student p2;

 p2 = p1;

 cout << "roll no : " << p2.roll\_no << endl;

 cout << "name : " << p2.name << endl;

 cout << "phone number : " << p2.phone\_number << endl;

 return 0;

}

Pada program di atas student adalah nama variabel structure /varibel bentukan kita, dimana akan menjadi tipe data untuk varibel p1 dan p2.

student p1;

student p2;

Saat ini akan dikembangkan bahwa tipe data bentukan tersebut akan kita gunakan untuk beberapa murid, misal 100 murid. Berarti di dalam tipe data tersebut kita definiskan utnuk varibel array sejumlah 100. Berikut ini adalah contohnya:

#include <iostream>

#include <cstring>

using namespace std;

struct student

{

 int roll\_no;

 string name;

 int phone\_number;

};

int main(){

 struct student stud[5];

 int i;

 for(i=0; i<5; i++){ //taking values from user

 cout << "Student " << i + 1 << endl;

 cout << "Enter roll no" << endl;

 cin >> stud[i].roll\_no;

 cout << "Enter name" << endl;

 cin >> stud[i].name;

 cout << "Enter phone number" << endl;

 cin >> stud[i].phone\_number;

 }

 for(i=0; i<5; i++){ //printing values

 cout << "Student " << i + 1 << endl;

 cout << "Roll no : " << stud[i].roll\_no << endl;

 cout << "Name : " << stud[i].name << endl;

 cout << "Phone no : " << stud[i].phone\_number << endl;

 }

 return 0;

}

Pada program di atas student adalah nama variabel structure /varibel bentukan kita, dimana akan menjadi tipe data untuk varibel array stud dengan indeks 5

student studs[5];

Demikian penjelasan:

* Array of function
* Pengenalan Struct
* Struct of function
* Array of function

Selanjutnya silakn diskusi di forum dan selanjutnya dibangun sebuah contoh sederhana untuk masing-masing individu mengenai pembahasan di atas. Seluruh mahasiswa wajib aktif di forum sebagai monitoring pemahaman topik dalam mingguan. Disarankan agar hasil program adalah running dari c++. Berikut outpunya juga screenshoot dari programnya yang dicopy ke word untuk dijadikan laporan tugas.

Masing-masing siswa bertanggung jawab terhadap pemahaman konten dan bila ada yang belum dimengerti bisa dipahami dalam forum. Mahasiswa bisa saling menambahkan dalam pengayaan pengetahuan sendiri dengan aktif di forum. Dosen akan aktif di hari Rabu dan Sabtu, selanujutnya akan diambil penilaian di hari minggu malam.

Selamat bekerja, terus semangat dan selalu berdoa.

Sumber yang digunakan:

<https://www.programiz.com/cpp-programming>