GeeksforGeeks

A computer science portal for geeks

Practice GATE CS Placements Q&A GeeksQuiz

Google™ Custom Search

Welcome mohitingale5

Are array members deeply copied?

In C/C++, we can assign a struct (or class in C++ only) variable to another variable of same type. When we assign a struct variable to another, all members of the variable are copied to the other struct variable. But what happens when the structure contains pointer to dynamically allocated memory and what if it contains an array?

In the following C++ program, struct variable st1 contains pointer to dynamically allocated memory. When we assign st1 to st2, str pointer of st2 also start pointing to same memory location. This kind of copying is called Shallow Copy.

```
# include <iostream>
# include <string.h>

using namespace std;

struct test
{
  char *str;
};

int main()
{
  struct test st1, st2;

  st1.str = new char[20];
  strcpy(st1.str, "GeeksforGeeks");

  st2 = st1;

  st1.str[0] = 'X';
  st1.str[1] = 'Y';

  /* Since copy was shallow, both strings are same */
  cout << "st1's str = " << st1.str << endl;
  cout << "st2's str = " << st2.str << endl;

  return 0;
}
```

Run on IDE

Output:

```
st1’s str = XYeksforGeeks
st2’s str = XYeksforGeeks
```

Now, what about arrays? *The point to note is that the array members are not shallow copied, compiler automatically performs Deep Copy for array members.*. In the following program, struct test contains array member str[]. When we assign st1 to st2, st2 has a new copy of the array. So st2 is not changed when we change str[] of st1.

```
# include <iostream>
# include <string.h>
```

```
using namespace std;

struct test
{
  char str[20];
};

int main()
{
  struct test st1, st2;

  strcpy(st1.str, "GeeksforGeeks");

  st2 = st1;

  st1.str[0] = 'X';
  st1.str[1] = 'Y';

  /* Since copy was Deep, both arrays are different */
  cout << "st1's str = " << st1.str << endl;
  cout << "st2's str = " << st2.str << endl;

  return 0;
}
```

Run on IDE

Output:

st1’s str = XYeksforGeeks

st2’s str = GeeksforGeeks

Therefore, for C++ classes, we don’t need to write our own copy constructor and assignment operator for array members as the default behavior is Deep copy for arrays.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

GATE CS Corner   Company Wise Coding Practice

C/C++ Puzzles

Related Posts:

- Printing source code of a C program itself
- isalpha() and isdigit() functions in C with example
- Nested printf (printf inside printf) in C
- Assigning an integer to float and comparison in C/C++
- Counts of distinct consecutive sub-string of length two using C++ STL
- auto_ptr, unique_ptr, shared_ptr and weak_ptr
- Interesting Facts in C Programming
- Dangling, Void , Null and Wild Pointers

Logged in as **mohitingale5** (Logout)

3.6 Average Difficulty : **3.6/5.0**
Based on **16** vote(s)

Basic Easy Medium Hard Expert

🟩 Add to TODO List

🟩 Mark as DONE

Writing code in comment? Please usecode.geeksforgeeks.org generate link and share the link here.

Load Comments

Contact Us! About Us! Advertise with us! Privacy Policy