Function overloading and return type

In C++ and Java, functions can not be overloaded if they differ only in the return type.

For example, the following program C++ and Java programs fail in compilation.

C++ Program

```
#include<iostream>
int foo() {
    return 10;
}

char foo() {  // compiler error; new declaration of foo()
    return 'a';
}

int main()
{
    char x = foo();
    getchar();
    return 0;
}
```

Java Program

```
// filename Main.java
public class Main {
    public int foo() {
        return 10;
    }
    public char foo() { // compiler error: foo() is already defined
        return 'a';
    }
    public static void main(String args[])
```


Interview Experiences
Advanced Data Structures
Dynamic Programming
Greedy Algorithms
Backtracking
Pattern Searching
Divide & Conquer
Mathematical Algorithms
Recursion
Geometric Algorithms

```
        {
        }
}
```

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

Related Tpoics:

- Can virtual functions be inlined?
- How to pass a 2D array as a parameter in C?
- How to dynamically allocate a 2D array in C?
- Convert a floating point number to string in C
- Integer Promotions in C
- Print a long int in C using putchar() only
- Interesting Facts about Bitwise Operators in C
- Interesting Facts about Macros and Preprocessors in C

Writing code in comment? Please use **ideone.com** and share the link here.

Popular Posts

8 Comments GeeksforGeeks

Sort by Newest

Join the discussion…

surajsahu06 • 7 months ago

I assume that is function overloading since it has diff type of parameter .My qu

Two functions, which has diff type of parameter, diff return type will be conside

```
public class Header {

public int addTwoNumbers(int a, int b){

return a+b;

}

public double addTwoNumbers(double a, double b){

return a+b;

}

public static void main(String[] args){

double a = 10.10;
```

see more

1 • Reply • Share ›

Rahul • 10 months ago

Can anyone tell the exact reason behind this?

Reply • Share ›

NAVEEN PRAJAPATI ➚ Rahul • 10 months ago

the return type of functions is not a part of the mangled name which is identifying each function. The
No of arguments
Type of arguments &
Sequence of arguments
are the parameters which are used to generate the unique mangled na of these unique mangled names that compiler can understand which fu same(overloading).Hence..... i hope u have understood what i am sayi

4 Reply • Share ›

OM ➚ NAVEEN PRAJAPATI • 3 months ago

Perfect Answer thanks naveen

Reply • Share ›

andy qi ➚ NAVEEN PRAJAPATI • 8 months ago

I am curious about how do you know that ?

Reply • Share ›

Rakesh Gupta • a year ago

It is not mandatory to collect the return value of the functions in a function call (code,

```
 int fun();
char fun();
int main()
```

705 Subscribe

Recent Comments

Abhi You live US or India?

Google (Mountain View) interview · 38 minutes ago

Aman Hi, Why arent we checking for conditions...

Write a C program to Delete a Tree. · 1 hour ago

kzs please provide solution for the problem...

Backtracking | Set 2 (Rat in a Maze) · 1 hour ago

Sanjay Agarwal bool tree::Root_to_leaf_path_given_sum(tree...

Root to leaf path sum equal to a given number · 1 hour ago

GOPI GOPINATH @admin Highlight this sentence "We can easily...

Count trailing zeroes in factorial of a number · 1 hour ago

newCoder3006 If the array contains negative numbers also. We...

Find subarray with given sum · 2 hours ago

```
{
    fun();
}
```

Here it's not possible to predict which function would be called. Hence compile

Also during name mangling, only function parameters, their types and sequenc name, not the return type. Hence two functions, who differ in return type only, v invalid. Two functions cannot have same mangled name(internal representatic

 • Reply • Share ›

kartik • 3 years ago

This explanation given by Venki looks the appropriate reason for not allowing o in C++.

 • Reply • Share ›

raghu • 3 years ago

Actually,
We can override methods based on return type in Java, this concept is new in return types. But the return type should be a subclass of the superclass.
Now what i mean by this.,

if method A is the main method and returns an object of class A, i can override Object of Class B provided if class B is a subclass of Class A

 • Reply • Share ›

 Subscribe Add Disqus to your site