

Function Overloading with Examples in C++ | C++ Tutorials for Beginners #19

In this tutorial, we will discuss function overloading in C++

Function Overloading in C++

Function overloading is a process to make more than one function with the same name but different parameters, numbers, or sequence. An example program to explain function overloading is shown in Code Snippet 1.

```
int sum(float a, int b){
    cout<<"Using function with 2 arguments"<<endl;
    return a+b;
}

int sum(int a, int b, int c){
    cout<<"Using function with 3 arguments"<<endl;
    return a+b+c;
}
```

Code Snippet 1: Sum Function Overloading Example

As shown in Code Snippet 1, we have created two “sum” functions, the 1st “sum” function takes two arguments “int a”, “int b” and return the sum of those two variables; and the 2nd sum function is taking three arguments “int a”, “int b”, “int c” and return the sum of those three variables. Function call for these “sum” function is shown in Code Snippet 2.

```
int main(){
    cout<<"The sum of 3 and 6 is "<<sum(3,6)<<endl;
    cout<<"The sum of 3, 7 and 6 is "<<sum(3, 7, 6)<<endl;
    return 0;
}
```

Code Snippet 2: Sum Function Call

As shown in Code Snippet 2, we passed two arguments in the first function call and three arguments in the second function call. The output of the following program is shown in figure 1.

```
The sum of 3 and 6 is Using function with 2 arguments
9
The sum of 3, 7 and 6 is Using function with 3 arguments
16
```

Figure 1: Sum Function Output

As shown in Code Snippet 3, both the “sum” function runs fine and gives us the required output. The main thing to note here is that the name of the function can be the same but the data type and the sequence of arguments need to be different as shown in the example program otherwise program will not run.

Another example of function overloading is shown in Code Snippet 3.

```
// Calculate the volume of a cylinder
int volume(double r, int h){
    return(3.14 * r *r *h);
}

// Calculate the volume of a cube
int volume(int a){
    return (a * a * a);
}

// Rectangular box
int volume (int l, int b, int h){
    return (l*b*h);
}
```

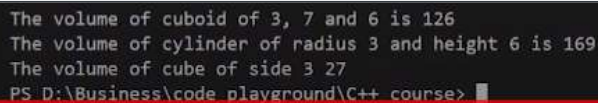
Code Snippet 3: Volume Function Overloading Example

As shown in Code Snippet 3, we have created three “volume” functions, the 1st “volume” function calculates the volume of the cylinder and has two arguments “double r” and “int h”; the 2nd “volume” function calculates the volume of the cube and has one argument “int a”; the 3rd “volume” function calculates the volume of the rectangular box and has three arguments “int l”, “int b” and “int h”. The function call for these “volumes” function is shown in Code Snippet 4.

```
int main(){
    cout<<"The volume of cuboid of 3, 7 and 6 is "<<volume(3, 7, 6)<<endl;
    cout<<"The volume of cylinder of radius 3 and height 6 is "<<volume(3, 6)<<endl;
    cout<<"The volume of cube of side 3 is "<<volume(3)<<endl;
    return 0;
}
```

Code Snippet 4: Volume Function Call

As shown in Code Snippet 4, we passed three arguments in the first function call, two arguments in the second function call, and one argument in the third function call. The output of the following program is shown in figure 2.



```
The volume of cuboid of 3, 7 and 6 is 126
The volume of cylinder of radius 3 and height 6 is 169
The volume of cube of side 3 is 27
PS D:\Business\code playground\C++ course>
```

Figure 2: Volume Function Output

As shown in figure 2, all three “volume” functions run fine and give us the required output.

Code as described/written in the video

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

int sum(float a, int b){
    cout<<"Using function with 2 arguments"<<endl;
    return a+b;
}

int sum(int a, int b, int c){
    cout<<"Using function with 3 arguments"<<endl;
    return a+b+c;
}

// Calculate the volume of a cylinder
int volume(double r, int h){
    return(3.14 * r *r *h);
}

// Calculate the volume of a cube
int volume(int a){
    return (a * a * a);
}

// Rectangular box
int volume (int l, int b, int h){
```

[< Previous](#)[Next >](#)