C++ Function Overloading

In C++ programming, two functions can have same name if number and/or type of arguments passed are different. These functions having different number or type (or both) of parameters are known as overloaded functions. For example:

- ✓ **`int test() { }`**
- ✓ **`int test(int a) { }`**
- ✓ **`float test(double a) { }`**
- ✓ **`int test(int a, double b) { }`**

Here, all 4 functions are overloaded functions because argument(s) passed to these functions are different.

Function Overloading Example:

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

void display(int);
void display(float);
void display(int, float);

int main() {

    int a = 5;
    float b = 5.5;

    display(a);
    display(b);
    display(a, b);

    return 0;
}

void display(int var) {
    cout << "Integer number: " << var << endl;
}

void display(float var) {
    cout << "Float number: " << var << endl;
}

void display(int var1, float var2) {
    cout << "Integer number: " << var1;
    cout << " and float number:" << var2;
}
```

Output

```
Integer number: 5
Float number: 5.5
Integer number: 5 and float number: 5.5
```

Here, the `display()` function is called three times with different type or number of arguments.