**Simple examples of a function template**

Templates are a very powerful feature of C++. They facilitate writing a single code segment for a set of related functions, called a function template, and for a set of related classes, called a class template.

Similar to variables being parameters to functions, types (that is, data types) are parameters to templates.

The following examples illustrate the use of a function template.

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

// Determining the larger of two data items
template <class Type>
Type larger(Type a, Type b); // Prototype declaration

int main()
{
    cout << "Line 1: Larger of 5 and 6 is: "
         << larger(5, 6) << endl; // Line 1

    cout << "Line 2: Larger of X and Y is: "
         << larger('X', 'Y') << endl; // Line 2

    cout << "Line 3: Larger of 5.75 and 3.25 is: "
         << larger(5.75, 3.25) << endl; // Line 3

    return 0;
} // main

template <class Type>
Type larger(Type a, Type b)
{
    if (a >= b)
        return a;
    else
        return b;
} // larger
```

**Output**

```
Line 1: Larger of 5 and 6 is: 6
Line 2: Larger of X and Y is: Y
Line 3: Larger of 5.75 and 3.25 is: 5.75
```
```cpp
#include <iostream>
using namespace std;

// Swapping values of two data items
// Type maybe double, int or char

template <class T>
void swap1(T& a, T& b);  // Prototype declaration

int main()
{
    double x, y;
    x = 2.5; y = 3.9;
    cout << "Before swap: x = " << x << " and y = " << y << endl;
    swap1(x, y);
    cout << "After swap: x = " << x << " and y = " << y << endl << endl;

    int i, j;
    i = 3; j = 7;
    cout << "Before swap: i = " << i << " and j = " << j << endl;
    swap1(i, j);
    cout << "After swap: i = " << i << " and j = " << j << endl << endl;

    char a, b;
    a = 'p'; b = 'q';
    cout << "Before swap: a = " << a << " and b = " << b << endl;
    swap1(a, b);
    cout << "Before swap: a = " << a << " and b = " << b << endl << endl;
    return 0;
}

template <class T>
// mandatory declaration before each function definition

void swap1(T& a, T& b)
{
    T temp;
    temp = a;
    a = b;
    b = temp;
}
```

**Output**

Before swap: x = 2.5 and y = 3.9
After swap: x = 3.9 and y = 2.5

Before swap: i = 3 and j = 7
After swap: i = 7 and j = 3

Before swap: a = p and b = q
After swap: a = q and y = p