**C++ Function Templates**

Function templates are another application of templates. Class templates were used to program generic classes. Similarly, function templates are used to define generic functions to be able to use a function independent of its data types for different data type combinations.

Example of some of the in-built function templates are,

sort(), max(), min() etc.

**The syntax for constructing a function template is,**

template <class T1, class T2>

data\_type function\_name(T1 a, T2 b)

{

//function body

}

An example of a somewhat data type independent function might be a function to find the average of two numerical values. Now, this function might expect any of the data type combinations between an integer, a float, or a double value. So, here is how we build a function so that it is generalized for finding the average of two numerical values.

#include <iostream>

using namespace std;

template <class T1, class T2>

float findAverage(T1 a, T2 b)

{

float avg = (a + b) / 2.0;

return avg;

}

int main()

{

float avg = findAverage(5.1, 2);

cout << avg << endl;

}