**Templates**

**Creating template tells the compiler that there is data type used in the class which will be specified afterwards.**

**Template with Multiple Parameters:-**

* We can specify more than one custom data type in a class by creating a template with multiple parameters. For e.g. if we want to use two arrays in the class – one of int type and one of float, we will create a template with two parameters.
* These custom data types can also be name of the class.

template <class T1, class T2>

class myClass{};

int main(){

myClass <int, float> obj(21,4.5);}

**Default Parameters:**

template<class T1=int, class T2=float, class T3=char>

class Harry{};

int main(){

Harry<> h1(21, 34.3, 'g'); //Here empty <> is necessary

}

**Function Template:**

Syntax:

template<class T1, class T2>

return\_type func\_name(T1 a, T2 b){}

* These can work as an alternative to overloaded function sometimes.
* Member function of a class can be defined outside a class with help of a scope resolution operator.

**Overloaded Template Function**

Exact match get highest priority.If exact match wasnot present then template function would have been invoked.