

The core idea of template is to be able to write a code which at the time of Compilation can generate further new code and that generated code get compiled again.

The main motivation of wing templates is an attempt to

consider a function, add (), which takes two numbers as irend parameters, adds there two numbers and returns the added value. If both the numbers are integers thes the weat for body of the function will be as follows:

int add ( int w, int y)

int ti 1. K+N = 7

neturn ti

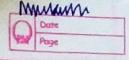
If both the numbers are floats then the body of the function will be as follows:
float add (float n, float T)

float ?;

y neturn til

In both the b functions, we can sente observe that in can of latter, the wind 'mi' has been replaced with 'flood'. 89, Otherwise, everything is some between both the functions. For such cares, CHA provides the concept of templates in which we no need to reparately provide unite two functions, one for int and one for fload. Here, only one function is written and the data type is passed as

a & parameter at the time of calling of the function. Fun example template (class T) Tadd (TXTY) E = M+Y', return ?; int major ) d int 2 = 2, 4 = 3; int 2; It here the data tipe t= add (int) (M,Y); ind in being cont LLt; parried as a parameter float a = 2.5, b = 3.5; flood C', c= add <fluat > (a,b); It has the data tipe 'fload' is being passes as paramety! contlec; In the above program, when int' is being passer as a parameter, in this can, the template parameter Tinke add () function will be neplaced by into and the function adde ) will behave in a such a way that it takes two integers as parameters and returns the sum of there two values integeter. Similar, when float is being parred as a parameter then the template parameter T with of the add () function will be replaced with the

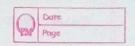


Similarly, consider a function man () which returns themanimum value among two variables. In care of two character variables, it returns a value with greater Ascss value. template (clan T) T man (Tx, TY) (f(x)) return M; else return y; int maine) int 1=2, 4=3 6, 2. 7 = man (int) (not) con (12) float a = 2.5, b = 3.5, (; ( = man < float > (a,b); chande (out ccc chand = 'm', e = 'H', f', f= man (chan) (d,e); Cost LLf; function overloading in demplate Function template with multiple parameters: 9 can of tempete parameters in function, we know that there can be be more than one parameter with different data type. es func int n, flood y) Similarly, in can of function template also, there can be multiple template parameters.



template (class Ti, class Tr) void funct, x, Try) cont cc "double template"; 2 Sun & cint, float > (5,6.5); Int main () 2) go gran of passing a simple parameter, we can remove the angular bracket while calling the function. es. come template ( Class T) void fun (T n) { coulce template; int maine) fun (5); / hem < int) By in games montiple temptate parameters, in water to write the angular bracket, But her in this care, in care of multiple parameters wish single template paraneles, the data types of the parameters being passed must be some otherwise we need to use the angular bracket for to internal type Conversion. R.S. template (clan T) void fun (Tx, T)) & conternery; main()

fun(5,5): 1' it will print 550



fun(5,6.5); It will give enough boot data types of

Parameters must be same as we used

Only single template parameters

we are not using anywlar bracked

while calling the function.

fun 6 (int 7 (5,6.5); /4 it will point 5,6

here, 6.5 will touncated to
6 since we are wring kint ).

Function overloading in template

As we know that, primary condition for function overloading is
the functions must be same.

Similar, in case of template also, above rules must be followed. Apart forom that, in a program, both functions with normal parameters as well as template parameters may exists, forming the situation of function overloading

Void fun (TM); (function with template

Void fun (int )); (func with normal

parameter)

In such situation, if a function Call enactly matches with the function with normal paradocates them this function will be pinvoked and not the function with template paramete.

In care of function with multiple parameters having single template parameter, implicit type convenion will be done with function with numal-parameter for function over loading.

to template ( class T > void fun (T n, Ty) Page Page outce "single temp" (LINCCY) template ( Class T1, Class T2) void fun (TIP, T2 2) (out LL "multi temp" ((PLCQ) void fun ( int &m, int n) cout ( "normal" ( ( m < ( h ) int main () fun (5,5); 1+ normal 5 50) fun (6.5,65); [ Single temp 6.565) , fun (5,6.5); /+ multi temp 5 6.5+1 template ( class T) 5 void fun( Tr, Tr) cond CC" Single temp LL NCLY; templated ( I tri , m tri ) and biev 2 cont ( " normal I'm con",

int main () 1+ normal 10 10+1 jun(10,10); fun (9.5, 9.5); 1 \* Single temp 9.59.5 \* 1 fun (9.5, 10); 1+ normal 9 10+1 here, & implicite type conversion will be done with the function with normal parameters. =) As of now, we learned how to use template in a function As we know that, a class to consists of member function as well as data member. So, in order to use the term concept of template in a class, we not only need to templatize he member functions but also the data member For enample, clan ab c int No flood y Public: void Setdetail (int m, and n) J= N; int square x () gretur X \* N; froat squar - y () { notion year;

If we want to use the concept of template in the class and then the class mar look as follows: Date template ( class TI, class T2) class abo Public: void Seldetail (TIm, T2n) ) A= N; TI Square-M () godin N# " Seretury 7+7; in main () abckint, float > m; m. Seldetail (5, 6.5); int P; P= m. square\_x(); pos. flood q; /\* 25 will be printed! q = m. square\_y (); cont LLq.