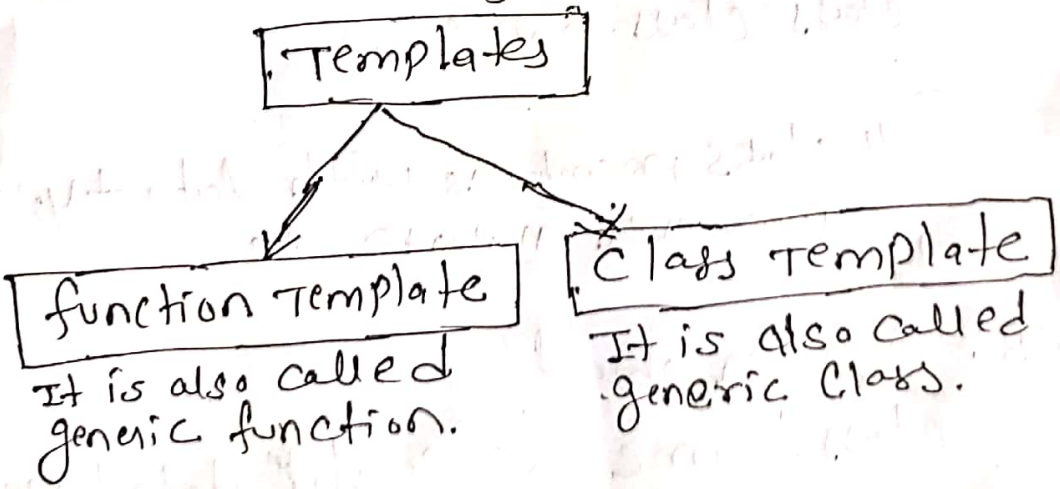
Template

In C++, a template is a powerful feature that allows functions and classes to operate with generic types. It means template allows you to write a single function or class that works with any data type without needing to write multiple versions of functions and classes for each data type.

Templates are especially useful for code reuse and for writing generic, type-independent algorithms or data structures.

"The generic family of functions or classes is known as template".

There are two types of templates in C++



function Template Contains two steps:

1. Defining general data type

template <class template_name>

② Defining a function with general data type

```
return-type function-name (parameters-with-
                           template-name)
{
  // body of function with type template-name
     whenever appropriate
}
```

Class Template Contains Three steps:-

step1: Defining template

```
template <class T>
```

step2: Defining class with members of template type

```
class class-name
{
  // class members with data types T
     whenever appropriate
};
```

step3: Defining object

```
class-name <specific-data-type> object-name
```

function Template (function Template is a generic funct

function template is used to define a function to handle arguments of different types in different times. we can use same function for arguments of one data type in some cases and arguments of other types in other cases.

for example we can define a function which can be used to add two integer or floating point numbers or double numbers.

Syntax:

```
template <class template_name>
return_type function_name(t a, t b)
{
  // body of the function
}
```

Example program

```
#include<iostream>
using namespace std;
template<class t>
void myfun(t a, t b)
{ cout<<"\n the value of a is: "<<a;
  cout<<"\n the value of b is: "<<b;
};
int main()
{ int p=10, q=20;
  float r=10.5, s=20.5;
  char c='a', d='b';
```

```
myfun(p,q);
myfun(r,s);
myfun(e,d);
return 0;
}
```

when compiler encounters a call to such general functions, it identifies the data type used in actual arguments and implements the function for that data type.

The advantages of function template is:

→ it avoids overhead of rewriting a function having body of same patter but operating for different data types.