TEMPLATE (genric classes)

TYPES OF TEMPLATE

1. FUNCTION TEMPLATE
2. CLASS TEMPLATE
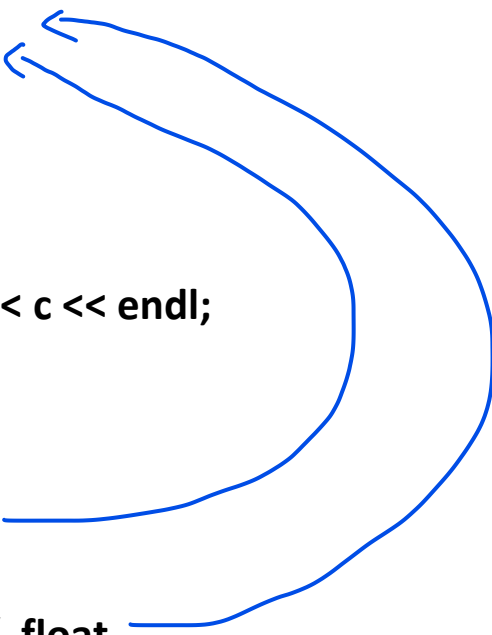
1. FUNCTION TEMPLATE

ADDITION OF TWO NOS

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

template < class t >
void    sum ( t  a , t  b )
{
        t  c ;

        c = a + b;
        cout<< " SUM = " << c << endl;
}
int     main()
{
        sum( 2,5 );    // int

        sum(4.3f , 2.8f );  //  float
}
```

FIND MAX NO.

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

template <class t>    // t --> type

t    max1 ( t  a , t   b )
{
    return( (a>b) ? a : b );
}

int    main()
{
    cout << max1( 3 , 2) << endl;          // 3

    cout << max1( 2.4f, 6.4f) << endl; // 6.4

    cout << max1( 'A', 'a')  << endl;    //   a

}
```

FUNCTION TEMPLATE WITH MULTIPLE AGRUMENTS

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

template < class  t , class  u >

void    write ( t  a , u b)
{
        cout<< a << endl;
        cout<< b << endl;
}
int    main()
{
      write ( 2 , 4.5 );

      write ( 2 , 'a' );

       write ( 2.3, 6 );

       write ( 5.6,'b');

       write ( 'a' ,4.3 );

       write ( 'c', 5 );

}
```

OVERLADING FUNCTION TEMPLATE

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

template <class t >
void   write( t a )
 {
       cout << a << endl;
 }

template <class t >
void   write( t  a  , int  n )
 {
       for( int  i = 1 ; i < = n ; i++ )
        {
               cout << a << endl;
        }
}
int    main()
{
      write(2);
      write('a');
      write(3.5);
      write(5.6,3);
      write('a',4);
      write(3,5);
}
```

CLASS TEMPLATE
INPUT AND PRINT TWO NOS

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

template < class t >
class  test
{
      private : t  a , b ;

      public  : void   get()
              {
                      cout<< " enter two nos " << endl;
                      cin >> a >> b;
              }
              void   out()
              {       cout<< a << endl << b << endl;
              }
 };
 int    main()
 {
      test <int> p  ;
      p . get();
      p . out(); // 3 , 2
      test <float>  q ;
      q . get();
      q . out(); // 3.4 , 5.5
 }
```

CLASS TEMPLATE WITH MULTIPLE AGRUMENT

INPUT AND PRINT TWO DIFF. NOS

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

 template < class t , class u >
 class   test
 {
      private :    t  a;
                   u  b;
      public :
          void   get()
          {
              cout<< " ENTER TWO NOS " << endl;

              cin >> a>> b;
          }

          void   out()
          {
              cout<< a << endl << b << endl;
          }
 };
    int   main()
    {
```

```
test < int , float >   p;

p . get();

p . out(); // 3 , 2.5

test < float , char > q;

q . get();

q . out(); // 3.4 , a


}
```

CLASS TEMPLATE

INPUT AND divide TWO NOS
```
#include<iostream>
using namespace std;

  template < class t >
  class test
```

```cpp
{
    private : t   a,b,c;

    public  :
            void   get()
            {
              cout<< " ENTER TWO NOS " << endl;

              cin >> a >> b;
            }

            void   out()
            {

              c=a/b;
              cout<<"value of c "<<c << endl;
            }
};


    int main()
    {

        test <int> p ;

        p . get();

        p . out(); // 3 , 2
```

```
test <float> q ;

q . get();

q . out(); // 3.4 , 5.5

}
```