Subject: Re: Nested template question
Posted by slashupp on Mon, 12 Aug 2019 07:56:18 GMT
View Forum Message <> Reply to Message

Don't know if I understand the problem correctly,
but I came up with the following:

```
#include <iostream>
#include <complex>

template<typename...P> void say(P...p){ (std::cout << ... << p); }

template<typename T> T mkZero(T &t) { t=T(0); return t; }
template<typename T, typename...V> T mkVal(T &t, V...v) { t=T(v...); return t; }

int main()
{
    int i;
    float f;
    double d;
    std::complex<int> ci;
    std::complex<float> cf;
    std::complex<double> cd;

    mkVal(i, 11);
    mkVal(f, 22.22);
    mkVal(d, 33);
    mkVal(ci, 44, 44);
    mkVal(cf, 5.5, 55.55);
    mkVal(cd, 66, 66.6);

    say("valued: i=", i, ", f=", f, ", d=", d, ", ci=", ci, ", cf=", cf, ", cd=", cd, "\n");

    mkZero(i);
    mkZero(f);
    mkZero(d);
    mkZero(ci);
    mkZero(cf);
    mkZero(cd);

    say("zeroed: i=", i, ", f=", f, ", d=", d, ", ci=", ci, ", cf=", cf, ", cd=", cd, "\n");

    return 0;
}
```

Output:

valued: i=11, f=22.22, d=33, ci=(44,44), cf=(5.5,55.55), cd=(66,66.6)
zeroed: i=0, f=0, d=0, ci=(0,0), cf=(0,0), cd=(0,0)