Arguments over Arguments

you already know this…

@adishavit :: #corecpp2019

```
#include <iostream>
#include "argh.h“

int main(int, char* argv[])
{
    argh::parser cmdl(argv);
    std::cout << cmdl[0];
}
```

```
>> myapp –v
```

```
C:\corecpp2019\myapp.exe
```

```
#include <iostream>
#include "argh.h“

int main(int, char* argv[])
{
    argh::parser cmdl(argv);
    std::cout << cmdl[1];
}
```

```
>> myapp Bach --verbose
```

```
Bach
```

```
>> myapp --verbose Bach
```

```
Bach
```

```
#include <iostream>
#include "argh.h“

int main(int, char* argv[])
{
    argh::parser cmdl(argv);
    for (auto arg : cmdl)
        std::cout << arg << ' ';
}
```

```
>> myapp Bach Mozart Chopin
```

```
Bach Mozart Chopin
```

```
>> myapp 3 2 1 0 -1 -two
```

```
3 2 1 0 -1
```

```
#include <iostream>
#include "argh.h“

int main(int, char* argv[])
{
    argh::parser cmdl(argv);
    int val = -1;
    cmdl(1) >> val;
    std::cout << val;
}
```

```
>> myapp 42
```

```
42
```

```
#include <iostream>
#include "argh.h“

int main(int, char* argv[])
{
    argh::parser cmdl(argv);
    int val = -1;
    cmdl(42, 2019) >> val;
    std::cout << val;
}
```

```
>> myapp 42
```

```
2019
```

```
#include <iostream>
#include "argh.h“

int main(int, char* argv[])
{
    argh::parser cmdl(argv);

    if (cmdl["-v"])
        std::cout << "Verbose, I am.\n";
}
```

```
>> myapp -v
```

```
Verbose, I am.
```

```
>> myapp --v
```

```
Verbose, I am.
```

```
#include <iostream>
#include "argh.h“

int main(int, char* argv[])
{
    argh::parser cmdl(argv);

    if (cmdl[{ "-v", "--verbose" }])
        std::cout << "Verbose, I am.\n";
}
```

```
>> myapp -v
```

```
Verbose, I am.
```

```
>> myapp --verbose
```

```
Verbose, I am.
```

```
#include <iostream>
#include "argh.h“

int main(int, char* argv[])
{
    argh::parser cmdl(argv);
    int answer = -1;
    cmdl("--answer") >> answer;
    std::cout << answer;
}
```

```
>> myapp --answer=42
```

```
42
```

```
#include <iostream>
#include "argh.h“
using namespace argh;
int main(int, char* argv[])
{
    parser cmdl(argv, PREFER_PARAM_FOR_UNREG_OPTION);
    int answer = -1;
    cmdl("--answer”, 2019) >> answer;
    std::cout << answer;
}
```

```
>> myapp --answer 42
```

```
42
```

```
>> myapp --question 42
```

```
2019
```

```
#include <iostream>
#include "argh.h“

int main(int, char* argv[])
{
    argh::parser cmdl(argv);
    float t;
    if (cmdl({ "-t", "--threshold"}) >> t)
        cout << t;
    else
        cout << cmdl("t").str() << “ !!!";
}
```

```
>> myapp -t=2019
```

```
2019
```

```
>> myapp -threshold=CoreCpp2019
```

```
CoreCpp2019 !!!
```

github.com/adishavit/argh

- **Header only**
- **C++11**
- **BSD**
- **No non-`std` deps**
- **No exception usage**
- **CMake+Buck builds**

Argh!

less is more

*** No automatic usage generation – bonus slide…**

```
// magic...

int main(int argc, char**)
{
   Foo::usage();
   return EXIT_SUCCESS;

   foo<1>();
   if (argc < 12)
      foo<2>();
   else
      foo<3>();
}
```

```
>> myapp
```

```
1 2 3
```

wandbox.org/permlink/29qV9

```
// C++20 magic...

int main(int argc, char**)
{
   argh::parser cmdl(argv);

    if (cmdl<“Print usage">["--help"])
        std::cout << cmdl.usage();

    if (cmdl<"Verbose output">[{"v“, “verbose"]))
        std::cout << "Verbose, I am.\n";
}
```

```
>> myapp --help
```

```
--help           Print usage
-v, --verbose    Verbose output
```