Command Line Arguments

Command Line Arguments

Examples: Why Command Line Arguments

Command Line Arguments

Examples: Why Command Line Arguments

- `apt-get install vlc`
- `git commit -a -m "correction of problem 3!"`
- `cp [file] [target path]`

Command Line Arguments in C and C++

Command Line Arguments in C and C++

```cpp
#include <iostream>
// argc is counter for the arguments, including the application-name.
// argv is array of strings representing the arguments.
int main( int argc, char *argv[] )
{
    std::cout << "Arguments count:" << argc << std::endl;

    for( int i = 0 ; i < argc ; ++i )
    {
        std::cout << "Argument:" << argv[ i ] << std::endl;
    }
}
```

Example: Simple calculater from command line arguments

Example: Simple calculater from command line arguments

What if we have a simple calculator like this.

Example: Simple calculater from command line arguments

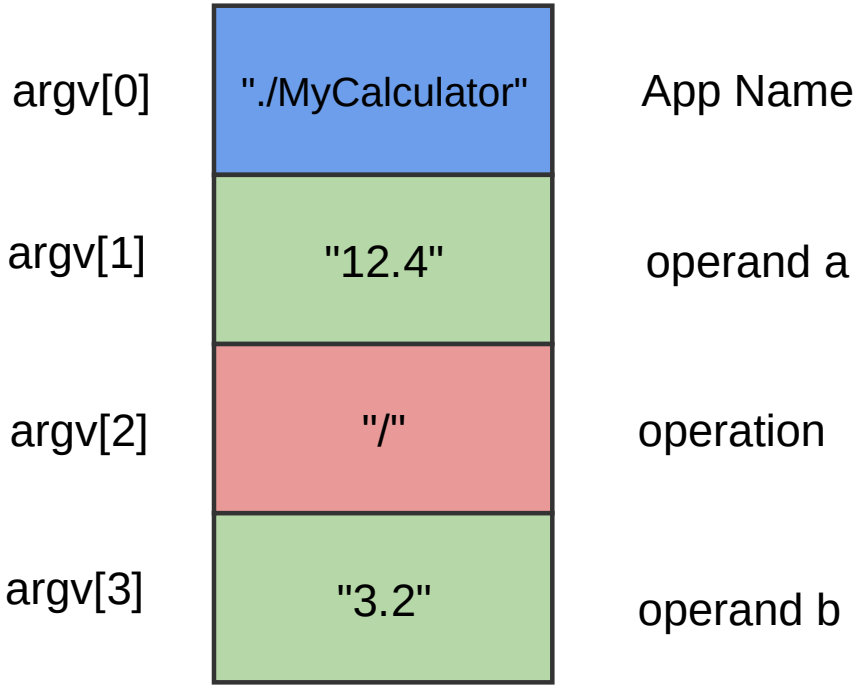
What if we have a simple calculator like this.

```
$ ./myCalculator 12.4 + 3.2
20
```

Example: Simple calculater from command line arguments

What if we have a simple calculator like this.

```
$ ./myCalculator 12.4 + 3.2
20
```



Convert from string form to integer form?

Convert from string form to integer form?

```
int x = "40"; // Compiler Error (Type Mismatch)

int y = std::atoi("40"); // Now this works, and y = 40.

double z = std::atof("13.9"); // z = 13.9
```

Convert from string form to integer form?

```
int x = "40"; // Compiler Error (Type Mismatch)

int y = std::atoi("40"); // Now this works, and y = 40.

double z = std::atof("13.9"); // z = 13.9
```

- `std::atoi` converts string representation => integer representation.

Convert from string form to integer form?

```
int x = "40"; // Compiler Error (Type Mismatch)

int y = std::atoi("40"); // Now this works, and y = 40.

double z = std::atof("13.9"); // z = 13.9
```

- `std::atoi` converts string representation => integer representation.
- `std::atof` converts a string representation => double representation.

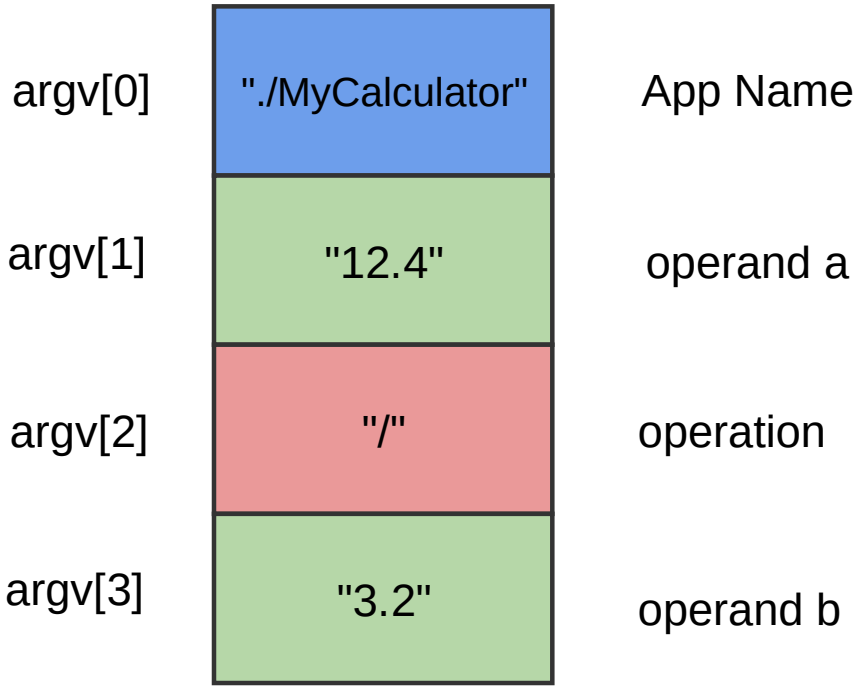
Convert from string form to integer form?

```
int x = "40"; // Compiler Error (Type Mismatch)

int y = std::atoi("40"); // Now this works, and y = 40.

double z = std::atof("13.9"); // z = 13.9
```

- `std::atoi` converts string representation => integer representation.
- `std::atof` converts a string representation => double representation.
- `#include <cstdlib>`.

```
#include <iostream>
#include <cstdlib>

// Our logic
double calculation( double a , double b , char operation );

int main( int argc , char *argv[] )
{
    double a = std::atof( argv[1] );
    double b = std::atof( argv[3] );

    char *op_string = argv[2];
    char op = op_string[0];

    std::cout << calculcation( a , b , op ) << std::endl;
    return 0;
}
```

```
double calculation( double a , double b , char operation )
{
  switch( operation )
  {
    case '+': return a + b;
    case '-': return a - b;
    case 'x': return a * b;
    case '/': return a / b;
    default:  return 0;
  }
}
```

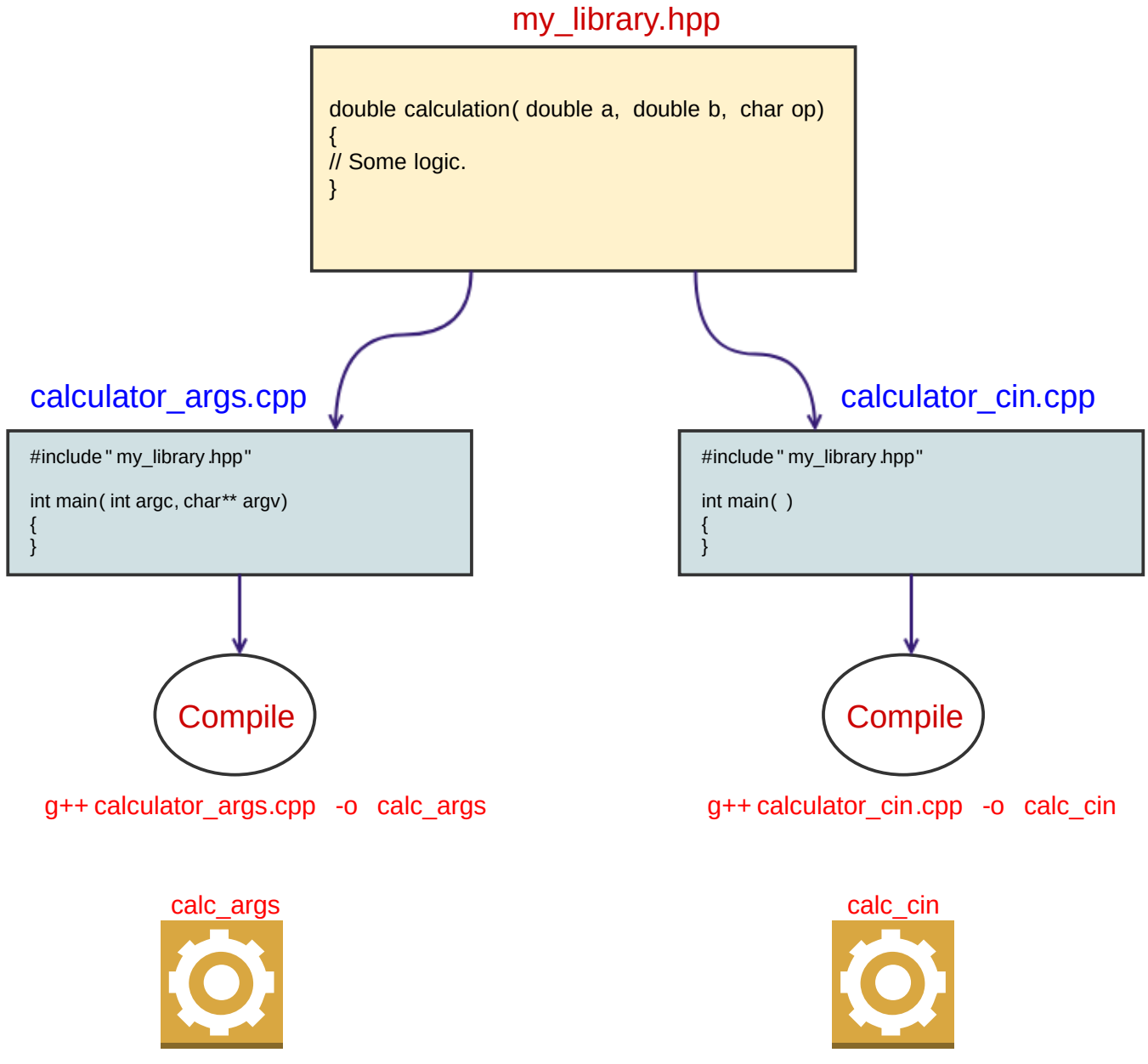

argv[0]
"./MyCalculator"
App Name
argv[1]
"12.4"
operand a
argv[2]
"/"
operation
argv[3]
"3.2"
operand b

Making Your Own Header Libraries

{DRY}

i.e Don't repeat your self

- Don't copy codes.
- If you need to fix a mistake, fix it from one place.


my_library.hpp
double calculation(double a, double b, char op)
{
// Some logic.
}
calculator_args.cpp
#include " my_library.hpp"
int main(int argc, char** argv)
{
}
calculator_cin.cpp
#include " my_library.hpp"
int main( )
{
}
Compile
Compile
g++ calculator_args.cpp -o calc_args
g++ calculator_cin.cpp -o calc_cin
calc_args
calc_cin