Regular: Yuqin Cui (s3215040)
Regular: Jaap van der Leest
Attendee: Timon van der Berg (s1925873)
October 2, 2017

C++ Course
Assignment 3

Exercise 18

Exercise 19

Exercise 20

Problem statement. Bla
Solution. main.cc:

```cpp
#include "head.ih"

int main(int argc, char* argv[])
{
    if (isatty(0))
    {
        std::cout << "no file redirection" << '\n';
        return 1;
    }

    std::cout << "PROCESSING" << '\n';
    process(arguments(argc,argv));
}

// -h and --help: provide short usage information and quit, returning 0 to the operating system. The usage info sho
// -v and --version: show the program's version and quit, returning 0 to the operating system;
// -c, --uc, and --captitalize: all letters in the input file should be capitalized;
// -l, --lc, and --lower-case: all letters in the input file should be transformed to lower-case characters;
// program expects a file, redirected to stdin; i.e. just cin
```

head.ih:

```cpp
#include <unistd.h>         // isatty
#include <iostream>         // cin, cout
#include <getopt.h>         // getopt_long

// processing type
enum class Mode {
    ERROR,
    OK,
    CAPITALIZE,
    LOWER_CASE,
    VERSION,
    USAGE
};

// arguments type
struct vars_t {
    bool help;              // -h --help
    bool version;           // -v --version
    bool capitalize;        // -c --uc --capitalize
    bool lowercase;         // -l --lc --lower-case
};

// info for user
void usage();

// process input
void process(vars_t Vars);

// do stuff
vars_t arguments(int argc, char* argv[]);

// select mode from arguments
Mode selectOpt(vars_t Vars);

// cout version num
void version();
```

arguments.cc:

```
#include "head.ih"

// long options and short options
struct option longOpts[] =
{
    {"capitalize",  0,    0, 'c'},
    {"lowercase",   0,    0, 'l'},
    {"version",     0,    0, 'v'},
    {"help",        0,    0, 'h'},
    { 0 }
};

vars_t arguments(int argc, char* argv[])
{
    vars_t Vars = {false, false, false, false};
    int opt;
    while ((opt = getopt_long(argc, argv, "hvcl", longOpts, &opt)) != -1)
        switch (opt)
        {
            case 'h':                   // help
            {
                Vars.help = true;
                std::cout << "HELP TRUE" << '\n';
                break;
            }
            case 'v':                   // version
            {
                Vars.version = true;
                std::cout << "VERSION TRUE" << '\n';
                break;
            }
            case 'c':                   // capitalize
            {
                Vars.capitalize = true;
                std::cout << "CAPITALIZE TRUE" << '\n';
                break;
            }
            case 'l':                   // lower-case
            {
                Vars.lowercase = true;
                std::cout << "LOWERCASE TRUE" << '\n';
                break;
            }
            default:
            {
                Vars.help = true;
                std::cout << "DEFAULT HELP TRUE" << '\n';
                break;
            }
        }
    return Vars;
}
```

process.cc:

```
#include "head.ih"
#include <ctype.h>          // toupper, tolower

void process(vars_t Vars)
{
    Mode option = selectOpt(Vars);
    switch (option)
    {
        case (Mode::ERROR):
        {
            std::cout << "ERROR" << '\n';
            break;
        }
        case (Mode::USAGE):
        {
            usage();
            break;
        }
        case (Mode::VERSION):
        {
            version();
            break;
        }
        case (Mode::OK):                                 // what is this mode for?
        {
            std::cout << "OK" << '\n';
            break;
        }
        case (Mode::CAPITALIZE):
        {
            std::cout << "CAPITALIZE" << '\n';
            break;
        }
        case (Mode::LOWER_CASE):
        {
            std::cout << "LOWER_CASE" << '\n';
            break;
        }
    }
    return;
}
```

selectopt.cc:

```
#include "head.ih”

Mode selectOpt(vars_t Vars)
{
    if (Vars.help)
    {
        return Mode::USAGE;
        std::cout << "SET MODE HELP” << '\n';
    }
    if (Vars.version)
    {
        return Mode::VERSION;
        std::cout << "SET MODE VERSION” << '\n';
    }
    if (Vars.capitalize and Vars.lowercase)
    {
        return Mode::ERROR;
        std::cout << "SET MODE ERROR” << '\n';
    }
    if (Vars.capitalize)
    {
        return Mode::CAPITALIZE;
    }
    if (Vars.lowercase)
    {
        return Mode::LOWER_CASE;
    }
    return Mode::ERROR;
}
```

usage.cc:

```
// instructions for users
#include "head.ih”

char const use[]=
    R”(
        <program's base name> V <version number>

        Usage: <program's base name> [options] < file
        Where:
            --captitalize (--uc, -u);   captitalize the letters in 'file'
            <... other options, alphabetically ordered>

        <program's base name processes 'file' and writes the results
        to the standard output stream.
        )”;

void usage()
{
    std::cout << use << '\n';
}
```

version.cc:

```
#include "head.ih”

void version(){
    std::cout << ”Version 1.45.12c.EY RC 5” << '\n';
}
```

Exercise 21

Problem statement.

Exercise 22

Exercise 23