

---

## EEE2007: Computer Systems and Microprocessors

Lab 6: Parameterizing C/C++ Programs

Module Instructor: Dr Rishad Shafik

---

### Exercise I: Making C++ Programs Parameterizable

---

**Recommended Time:** 35 Mins Maximum

**Aims:**

- a. To gain valuable working experience in parameterized C++ programs
- b. To understand the usage of strings

Follow the instructions below and try to do accordingly-

1. **DOWNLOAD** the source code of example1.cpp.
2. **REVIEW** the source code of example1.cpp using Notepad++ (Start->type "Notepad++")

Go through each line to understand how the code is organized. Check the following:

- How string classes are used to create powerful character array types
- How argc counts the number of arguments from main
- How argv stores the C character arrays of the input parameters
- How C++ string is used to compare with the desired string
- How stringstream is used to parse numbers from the strings

3. **COMPILE** the source code of example1.cpp:
  - a. Start Cygwin command shell through Start->All Programs->Cygwin->Cygwin Bash Shell
  - b. In the Cygwin shell type: `g++ -Wall example1.cpp -o example1`

The `-Wall` option enables all the warnings, and the `-o` option enables specification of the output executable

Your compilation should generate an executable called *example1*

4. **EXECUTE** the *example1* executable by typing the following in the Cygwin shell  
`./example1 -l <num1> -h <num2>`
5. **OBSERVE** the outputs:
  - a. `./example1 -l 2 -h 2324`
  - b. `./example1 -l 2324 -h 2`
  - c. see how parameters are processed and validated

**QUESTION:** Can you try with more values and improve the program further by using variable types that allow larger number ranges?

## Exercise II: Operating with Files

---

**Recommended Time:** 25 Mins Maximum

**Aims:**

a. To understand how files can be used to store large outputs

Follow the instructions below and try to do accordingly-

9. **DOWNLOAD** the source code of `example2.cpp`.

10. **REVIEW** the source code of `example2.cpp` using Notepad++ (Start->type "Notepad++")

Go the new lines in the original program (`example1.cpp`) and try to understand the program. Check the following (you can do "`diff example2.cpp example1.cpp`" to find the differences:

- How files are being opened and manipulated
- Why validation is necessary for opening files
- Note the file location can be specified relatively or by specific addressing methods

11. **COMPILE** the source code of `example2.cpp` by

- a. Start Cygwin command shell through Start->All Programs->Cygwin->Cygwin Bash Shell
- b. In the Cygwin shell type: `g++ -Wall example2.cpp -o example2`

Your compilation should generate an executable called *example2*

12. **EXECUTE** the *example2* executable by  
`./example2 -l <num1> -h <num2>`

13. **OBSERVE** the output with different inputs

## Exercise III: Operating with Files using parameters

---

**Recommended Time:** 30 Mins Maximum

Modify `example2.cpp` and add another pair of parameters in the list of arguments "`-f <file_name>`", where "`-f`" is the keyword to suggest that the following "`file_name`" will be used as the output file, where the range of numbers and the prime number lists will be printed.

For example,

```
./example1 -l 2 -h 2324 -f results.csv
```

Should take

- the low range as 2
- the high range as 2324
- the output file name as results.csv