

Concordia University
Department of Computer Science and Software Engineering

SOEN341 Fall 2020 - Project A - Part 1

Purpose

The main objective of this project is to get students' attention at the beginning of the course by means of enjoyable and short individual programming development activities as well as to engage them in the steps that followed: the main course project (B). Students will learn and apply appropriate warm-up “Études” (more details in lecture times) such as one or two (per topic):

- naming conventions & EBNF notation,
- clean-code practices,
- SOLID principles,
- idioms (programming patterns),
- basic refactoring techniques,
- basic essential design patterns, and
- Agile practices.

These warm-up activities increase students' attention and help to link the Agile software processes to learn in class.

Background, Context, and Tasks

The original UNIX paper¹ describes the notion of shells, filters, I/O redirection and pipes. The Unix system had a significant impact on other operating systems. Unix's innovative command-line syntax for creating modular chains of producer-consumer programs (pipelines) made a powerful programming paradigm. Many later command-line interpreters have been inspired by the Unix shell.

In a small company, several small and useful software tools as command-line programs (also called "batch" programs) were all designed in a traditional and procedural way. They are hard to maintain and impossible to (binary) reuse in whole or in part when trying recently to build new software tools.

So, over the next three weeks, the CEO has decided to make a first quick agile transition to his software developers. This transition will generate some short-term wins with a project to anchor new software development practices part of an agile process. The main task of this project (called **Project A**) is to do a complete object-oriented redesign in Java of some of his traditional software tools in the company which are the following command-line programs written in C:

C Programs	Short Description
copy.c	Copy sourceFile to destinationFile
charcount.c	Count characters in a sourcefile
linecount.c	Count lines in a sourcefile
wordcount.c	Count words in a sourcefile
wc.c	Count characters, lines, and words

Notation and Conventions to Follow

Extended Backus-Naur Form (EBNF) notation

This project will use the Extended Backus-Naur Form (EBNF) notation as summarized in Table 1.

Notation	Meaning
{ A }	Repetition—zero or more occurrences of A
A+ or A { A }	Repetition—one or more occurrences of A
A? or [A]	Option—zero or one occurrence of A
A B	Sequence—A followed by B
A B	Alternative—A or B
(A B)	Grouping—of an A B sequence

Table 1: Extended Backus-Naur Form (EBNF) notation

¹The UNIX TimeSharing System, Dennis M. Ritchie and Ken Thompson, Bell Laboratories, 1974

Conventions for Command-line Syntax

We use the above notation for defining a typical command-line syntax:

```
CommandLine = CommandName { Option } { Argument }
```

The following command-lines are examples of using the above C programs:

```
copy src.txt dst.txt
charcount copy.c
linecount copy.c
wordcount src.txt
wc src.txt
wc src.txt dst.txt
```

Roles

The following are the roles in this first project with a simplified Agile Process:

- **Domain Expert:** Your instructor will be your Domain Expert (DE), not always available :) which is often like the ones in the reality of industrial projects.
- **Customers:** Your TAs (An Ran Chen and Zhenhao Li) will be your Customers, Functional Analysts, Business Analysts, or Product Owners (POs) for developers. Developers could be lucky and have direct access to Domain Expert (during lecture times and office hours)
- **Students:** You as developers, apprentices, or learners of an agile software process. You will start wearing many hats: analyst, designer, programmer, tester, etc.

Tasks

The project A is planned for three weeks which are organized into iterations: One iteration (sprint) per week to achieve individual mini-tasks (reasonable targets):

Overview of sprints in Project A

Project A (3 Weeks / 3 Sprints):

- **Week 2: Sprint 1** - Rewrite C programs in Java, add an help option to all of them. Make sure to validate all option(s) and argument(s) passed to the command-line of each Java program and generate a usage message in case of errors.
- **Week 3: Sprint 2** - Design an administrator class to parse option(s) and argument(s) and redesign the `wc.c` program using classes for counters: `CharCounter`, `LineCounter`, and `WordCounter`. (more details on week 3)
- **Week 4: Sprint 3** - First team effort to improve your individual design and implementation in Java. (more details on week 4)

The help option has three representations that must be supported in all your corresponding Java programs:

```
Option = "-?" | "-h" | "-help"
```

where:

Short version	Long version	Meaning
-h or -?	-help	Print the usage of the program

Table 2: Three possible (and valid) representations for the help option.

In case of invalid options, you must print a usage message. The following are examples for the `copy`, `wordcount`, and `wc` programs:

```
C:\> copy
Usage: copy sourceFile destinationFile
C:\> wordcount
Usage: wordcount sourceFile
C:\> wc
Usage: wc sourceFile+
```

or

```
Usage: wc sourceFile { sourceFile }
```

Submission of Sprint 1 in Project A

For this sprint, **you must work individually. Only electronic submissions will be accepted.** Zip together the Java source codes only. Please use WINZIP. Naming convention for zip file is:

```
ProjectA_Sprint1_StudentID_StudentName.zip
```

Example:

```
ProjectA_Sprint1_12345678_JohnSmith.zip
```

Following your submission, a demo is required. The marker will inform you about demo times. Please notice that failing to demo your sprint will result in zero mark regardless of your submission.

Deadlines and Grading Scheme

Due dates for each Sprint:

- **Week 2:** Sprint 1 - Due 11:59 PM - Monday, September 21st, 2020
- **Week 3:** Sprint 2 - Due 11:59 PM - Monday, September 28st, 2020
- **Week 4:** Sprint 3 - Due 11:59 PM - Monday, October 5th, 2020

Total mark for Project A is 10%.

Each sprint will be evaluated as follows:

- **Sprint 1** - 3% - Evaluation criteria is based on your demo.
- **Sprint 2** - 3% - Evaluation criteria will be specified at the beginning of Sprint 2
- **Sprint 3** - 4% - Evaluation criteria will be specified at the beginning of Sprint 3

The first two sprints are 3% as individual marks. The remaining 4% for Sprint 3 is team marks. There is 1% for review contribution and peer-evaluation (more details in Week 4).

Your next tutorial

Your TAs, An Ran Chen (section SA and SB) and Zhenhao Li (section SC) will give you this week (week 2) a great tutorial to know how to use Git & GitHub.

Git is an open-source, distributed version control system designed for speed and efficiency. A version control system allows you to save a series of snapshots (commits) of your code.

GitHub is a web-based system on which you can remotely publish your Git repositories and collaborate with other people.

You will learn:

- A typical basic workflow of Git
 - Set up the user profile
 - Get the repository
 - Work on the files in the repo
 - Push your changes to a remote repo
- How to collaborate with others by using GitHub (more for Project B)
 - Creating a new repository
 - Branching and merging
 - Pulling request (for changes)
 - Reviewing comment, approve or request changes
 - Merging executed and pull request closed
- How to use Issues with GitHub in Projects (more for Project B)
 - Using milestones (Sprint) to collect and categorize the issues, etc.

Group Information

A major portion of the semester involves teamwork. Students are expected to produce teams of 6 to 7 maximum, no more students permitted per team unless a very special request is allowed.

One member per team has to email (Zhendao or An Ran) with the team members and their student IDs. **Please have this done by the end of week 3, which is Friday, September 25th.** If you do not have a team, e-mail An Ran so he (with Zhendao) can resolve the issue as fast as possible.

An Ran Chen: archen94@gmail.com

Zhenhao Li: lzh9410@gmail.com

There will be a code/design review on Week 4 (after producing teams).

Information included in your email:

1. Full name and student ID of each member
2. Which section you have registered (SA / SB / SC)

When done, we will assign all the students and post your group information on Moodle. **Please start finding your groups!**