

Dossier Checklist

This is a basic checklist that you may use to verify the completeness of your dossier. PLEASE be sure to cover each section's requirements and for more in-depth explanations of each section, visit Mr. Bui's website at: <http://paulbui.net/wl/Dossier> . If you have any questions, please e-mail him: paul_bui AT apsva DOT us

General Formatting Guidelines

- 12-point font. Times New Roman please
- Single-spaced (be sure to separate paragraphs by indenting or spacing)
- Cover/Title page (title, name, date, period)
- Table of Contents
- All pages are numbered in the lower right-hand corner (do not number the title page)
- Headings at the beginning of each of section

A1: Problem Analysis

/ 3

- Introduction
- Identify the specific end-user
- Problem statement (explicit)
- Problem discussion & description
- Alternative solutions to the problem
- Questions you would ask your end-user
- End-users requirements for the system
- List & discuss possible inputs
- List & discuss possible outputs
- Discuss possible interfaces/sub-programs
- Graphically illustrate possible user actions (flow chart of what users may do)

A2: Criteria for Success

/ 3

- Introduction (~2 sentences)
- List/outline the behavior of the program (things the program will let the user to do)
- List and describe the program's objectives (all things the program will do itself, usability, error-prevention, etc.)
- Specifically relate each objective back to some problem from the problem analysis

A3: Prototype Solution

/ 3

- User-action flow chart that is fully labeled
- Illustrations of the prototype system
- All illustrations have captions/explanations
- All illustrations correspond to a box in the user-action flowchart (label both so you can correspond the two)
- User feedback with your prototype design

B1: Data Structures

/ 3

- Subsections for each data structure
- Discuss and justify why the data structure was chosen
- Discuss alternative data structures for each
- Diagrams of what the structures look like
- Sample data in the data structure to help show what it looks like
- Illustrations showing the adding, removing, and updating of data structures

B2: Algorithms

/ 2

- Subsections for each algorithm used (searching, sorting, etc.). NO setters, NO getters
- List each algorithm's input parameter(s) and return value(s)
- Explain steps of algorithm in pseudo-code

B3: Modular Organization

/ 2

- Diagram the hierarchy of all your classes and how they connect to each other
- Subsections for each class
- Explain each class's purpose
- List and explain what all the major methods of each class does (no need to explain setters or getters)

C1: Code & Good Programming Style / 3

- PUT ALL YOUR CODE IN THIS SECTION**
- Commented header at the top of each class (program name, author, date, school, type of computer, Eclipse IDE, purpose of the file)
- Variables are named descriptively
- Good indentation
- Code is commented explaining how it works in most places

C2: Handling Errors / 3

- Table that contains the following:
- Brief descriptions of the errors detected
 - Descriptions of how the errors are resolved
 - References to your code that specifically shows you handling the errors

C3: Success of Program / 3

- Table that contains the following:
- Brief descriptions of all your objectives and goals taken **DIRECTLY** from your Criteria for Success
 - References to the screenshots illustrating that the objective was successfully met

D1: Test Output / 4

- Subsections for each test of your program. You should test every objective / goal listed in your Criteria for Success, Usability, and Handling Errors sections.
- Screenshots with captions / explanations

D2: Program Evaluation (Conclusion) / 4

- Discuss answers to the following:
- Did it work?
 - Which criteria from A2 were successfully met?
 - Which data sets did it work for?
 - How efficient is your program? (BIG-O!)
 - Does the program have any limitations?
 - What additional features would you add?
 - Was your initial design appropriate?
 - How would you design your program differently in the future?
 - What future enhancements could be made? (e.g. networking)

E: Holistic Approach / 2

Mastery Aspects / 10

- Table that contains the following:
- List of all the mastery aspects achieved
 - Descriptions of how / why each mastery aspect is achieved
 - References to the page(s) in the code that achieves the mastery aspect

Total / 35 x / 10 = / 35