Software Development Report
Replacing Pre– and Post–lab Exercises

1. **Problem Summary.**
   This is a short paragraph that briefly describes the problem that your software solves. This provides the context and scope for the reader.

2. **Specifications.**
   This is a list of things that any solution to the problem is required to do. This section answers the question “What?”, not “How?”. Include a list of required inputs and outputs, along with their types and ranges. If files are involved, describe the required or produced field formats.

3. **System Design and Detailed Design.**
   This is where “How?” is answered. Initially, this is your first design for the system. This section might go through a number of iterations. It might also change as the code is debugged. The final design must correspond with the developed code and should aid the reader in understanding and maintaining the code if necessary. This section should provide a roadmap for the software. The format of this section should use pseudocode to describe a general algorithm (break-down of your problem solution) followed by specific algorithms (for any large subproblems noted in the general algorithm).

4. **Testing Report.**
   This is a detailed description of your testing plan and the results of following that plan. It should include a description of major runtime errors encountered and the testing data used. If you claim your solution is correct, you need to justify that claim in this section. If your software has errors, you need to enumerate those errors and explain the cause as much as you are able. It is common to list some of the testing data and the results generated by your software in this section. You should include Exception, Boundary, and Normal testcases, along with Expected results and Output from the actual execution with those testcases.

5. **Management Report.**
   This is a report on how the development of the software was managed. It discusses how time was managed, what was done when, and what decisions about the process were made. You should indicate the order and time spent on design, coding, testing and debugging, and writing this report.

6. **Lesson Learned.**
   This section is where you explain what you learned doing this assignment. This might include a short discussion about the topic the assignment was about and additional items about programming that were learned in the process of the assignment.

7. **Future Improvements.**
   This is a short description of what could be done to improve your program in the future.

8. **Appendices (if needed).**
   Include a listing of your program unless otherwise instructed.

Over for instructions.
The instructions on the other side of this handout in *italics* should be replaced with your actual report, which should include the subtopics listed for each part.

**Software Development Report, Part I**

Starting with Lab 6, you are required to bring a **Software Development Report, Part I** for each lab exercise unless otherwise indicated. This may be handwritten as long as it is legible, and I may request you show it to me at any time during the lab. Each section must be labeled as follows.

**Name:** *(your name and user ID)*
**Assignment:** *(Lab # & Exercise #)*
**Date:** *(date of lab)*

1. **Specifications**

2. **System Design** — a general algorithm indicating order of processing, methods and arguments, and how the methods will be used

3. **Test Suite**

**Software Development Report, Part II**

The following week, you are required to produce the **Software Development Report, Part II**, which is to be typeset, for each exercise, unless otherwise indicated. Staple Parts I & II to the printout of each exercise. Each section must be labeled as follows.

**Name:** *(your name and user ID)*
**Assignment:** *(Lab # & Exercise #)*
**Date:** *(date turned in — next lab)*

1. **Testing Report**

2. **Management Report**

3. **Lesson Learned**