Laboratory Requirements

The laboratory part of this course is essential to the understanding of the course material, and so the following requirements have been established to ensure that everyone gets the most benefit from the laboratory.

Pre-lab

Because the time available during the scheduled laboratory period is limited, it is necessary to do a certain amount of preparation ahead of time. Many of the project instruction sheets will include a pre-lab section to be completed before coming to the lab period. These are intended to help you in preparation for the lab period, and the answers to questions asked are to be handed in at the beginning of the lab period.

Laboratory Sessions

Because of the limited number of PCs available, two or three persons will be working together as a group on each PC. However, each member of the group is required to hand in his/her own report. Discussion and co-working are only allowed within the group. All work turned in as a lab report must be an original creation of the person turning in the report. Copying of the pre-labs and reports is absolutely prohibited. If exact copies of work are turned in, the credit will be divided between the authors.

Report Format

The laboratory reports are to be as concise as possible, but the following components must be included.

1. Introduction: Summarize the problem, any specifications, and the objective. This should be self-contained (i.e. don’t write “read the handout”). Write in your own words, but be concise. Copying from the lab manual and the handout will not be accepted.

2. Software design: Include flow chart or pseudo code that describes your design. This is an outline of the solution approach that you use. It can serve you as a guide while writing the assembly language code for your solution, and it explains to others how you solved the problem. The flow chart may be hand drawn, but must be neat.

3. Test results: Give some examples that illustrate what your code does. For example, you can give the inputs to your software and the results that it generated.

4. Answers to the questions: If there are specific questions to be answered as part of the project, provide brief and clear answers.

5. Conclusion: Summarize briefly what you have accomplished, any problems that you encountered, and recommendations for improvement of the project.

6. Program listing: This includes your source code or the output of the assembler. You must document your program at the beginning of the source code. Include your name, student id, program file name, date, purpose of the program, and a short instruction on how to use the program. State the kinds of inputs required and the outputs expected. It is extremely important to put comments at appropriate places to explain what that program segment is doing.

The lab report must be typed and stapled together securely.

Programming Standards

The following general programming style standards are to be followed:

1) Module Header: Each source module (file) must begin with a comment block that introduces the module. This comment block should describe the basic purpose of the module, any special algorithms used in the module, any special data structures used or maintained by code in the
module and how it fits into the rest of the program. The module header should also identify the author of the module and have a revision history section.

2) Procedure Headers: Each procedure, function or subroutine must have a comment block at the beginning describing the function. This header block must include a description of all inputs used by the function and all outputs produced by the function. It should also include descriptions of any special algorithms or data structures used by the function and any error conditions detected by the procedure and what happens when errors occur.

3) Explanatory Comments: Each procedure must contain comments explaining what is happening and/or how the code works. These should describe the procedure in sufficient detail that someone reading the comments can understand how the procedure works with only minimal reference to the code.

Examples of recommended programming style are posted on the class web site. There are also template source modules provided which can be used as a starting point for creating new source file that will have a consistent style.
**Laboratory Grading**

- Pre-lab       20%
- Demonstration of program  30%
- Lab report     50%

Note: All pre-labs must be individual work. **Lab reports and demonstrations are due at the beginning of the following lab session.**

**Lab report grading scale**

Total number of points for a lab report is 50. Lab reports will be graded according to the following scale.

- **Introduction**
  - Specification of the lab
  - Objective of the lab
  
- **Software design**
  - flow chart or pseudo code
  - correctness
  - level of detail

- **Test results**
  - inputs and output

- **Answers to questions**
  
- **Conclusion**
  - lab summary
  - encountered problems and/or recommendations

- **Program listing**
  - conformation to format
    (All programs must begin with author(s) name, code, date, purpose of program)
  - comments

- **Total points:** 50