

Software 1 - Submission Guidelines

General

The following page describes the submission guidelines for the exercises in the course 'Software 1'. The content of the various exercises will be separately detailed. The exercises will be checked both by a person and by an automatic program. Therefore, it is very important to follow carefully the guidelines below. Failing to follow the exact names, directories, files, times, etc. might lead to the failure of the automatic checking mechanism, which in turn, will cause a significant reduction in the exercise grade.

Procedural Issues

- Each student must submit all the exercises by himself/herself. You may share ideas, but you may not share code lines! Severe actions will be taken against students whose work will be suspected as copied!
- Homework should be submitted on time. Students, who serve in the army during the semester, are asked to present an official confirmation and will receive additional time for submission. Failing to submit the homework in time due to any other reason is unacceptable and will be regarded as if not submitted.
- The exercises will be programs in Java 1.5 and should generate no warnings whatsoever.
- Deliverables - The exercises should be submitted both as jar files (one jar file per exercise) in your Linux directory, and as hardcopies (details are below).

Hardcopies

- The printouts should be placed in mail box of the exercise checker (see the course website for his/her details).
- The printouts you submit should be composed of nicely stapled papers. The first page should include the following:
 - Your full name
 - Your ID number.
 - Your user name (login).
 - Exercise sheet number.

Then, there should appear the source code of the tasks. Each task should start on a new page, and the number of the task should appear on the top (this may be added with a pen or a marker).

Files and Directories

- For each exercise you should create one jar file as explained in the Eclipse handouts. The names of the jar files should be 'ex*n*.jar', where *n* is the number of the exercise. For example, the name of the jar file for exercise 1 will be 'ex1.jar'.

- The jar file should be submitted on the nova UNIX machine. Specifically, each student should have a subdirectory under his/her home directory, called ~/software1 in which the jar files of all the exercises will be placed.
- You must not touch or change the files you submit after the deadline. Each file has a property of the last time it was changed. Files that are changed after the deadline will be considered as not submitted on time!
- The permissions of the entire path from your home directory down to the exercise files themselves should provide read and execute access for everyone. Details on how to set the permissions are in the [system handouts](#).

Questions and Support

Unclear issues or problems regarding the Linux environment, the nova machine, or any other system relating issues should be directed to the system consultants. They can be accessed via email (system@cs.tau.ac.il) or in person in the advisors room on the first floor in Schreiber building, room 019.

Other questions, regarding procedural or programming issues should be coped with in the following order:

1. Read the relevant exercise directions
2. Take a moment and think again whether the question makes sense and cannot be solved alone
3. Check the course's forum for previous answers or ask your question there
4. Questions about the checking procedures, grades, directories and files that were misplaced, etc. please send to tochna1@post.tau.ac.il (your exercise checker). If all else fails, send an email to the course mailing list stating the question shortly and clearly and one of the course's staff will answer it. Such an email should be in plain English only and its subject field should be Software 1. Students are also welcomed to answer questions sent to the mailing list.
5. Exercises should be read upon receiving them. Questions regarding the exercise should be submitted up to 5 days before the submission deadline, to allow sufficient time to answer.

Grading Criteria

The grades will be composed of the following parts:

1. **Submission on time.** Handing in printouts after the due date or making any changes in the files after that time are counted as not submitting at all. As a rule of thumb, there will always be an 'objective' problem in the day preceding the submission; a power failure, a printer malfunction, printer running out of paper, etc. Therefore, prepare your homework way in advance. Leave plenty of time for debugging, testing, and dealing with unexpected problems. Unexpected problems are the most expected things in programming projects!
2. **Correctness of the program.** The correctness will be checked by an automatic mechanism, and therefore a special care should be taken meeting the exact syntactic requirements.

3. **Design.** The programs should be well designed using the concepts studied in class (e.g., object-oriented, design by contract)
4. **Implementation efficiency.** Your programs are expected to be reasonably efficient.
5. **Documentation.** The inline documentation is composed mainly of a reasonable amount of comments (when needed). Make sure that each file and function includes a short paragraph explaining its purpose, IO and method of work.
6. **Readability and Clarity.** Naming - Use intelligent naming for variables, methods and classes. Modularity - divide code into methods and classes (a by product of good Object Oriented design). Comments.
7. **Indentation.** Sometimes a Tab is better than thousands words. Note that the Eclipse editor can do it automatically.
8. **Honesty.** No extra points will be given for honesty. Any suspicion of deception, however, will cause immediate and severe steps.