System Programming in C
Homework Submission Guidelines

This document contains important information regarding submission of homework assignments in the course. Please read this carefully before working on the first assignment. In particular, we direct you to the two sections on **our philosophy** and **sharing ideas** that convey our policy and expectations regarding homework in the course.

Good Luck!!

Ilan, Sara, Tuvia and Liam.

Contents

1. Course philosophy regarding homework

As with every programming course, homework assignments play a central role in the teaching experience. You cannot learn how to program just by watching the lectures and reviewing the slides. We put a lot of thought in developing the assignments, checking them and grading them, and we expect you to put effort in their solution. This is the best way for you to reach the end of the semester prepared for the final exam. The course moves very quickly through a diverse set of topics, and each of the six homework assignments in the course is designed to cover a different topic. This is why we encourage you to put thought into **every assignment** and why we do not exclude any assignment from the final grade – **no n-1 rule in the course**. We prefer that you put some thought into each of the six assignments rather than work hard to get 100% on five of them. On the other hand, we realize that you may not always have time to devote your fullest to each assignment. This is why the assignments count only for 20% of your final grade. Each assignment is worth less than 4 grade points and 10 points on an assignment are worth less than ½ a grade point in your final grade. So, if you do not have time to complete an assignment, submit a partial solution and receive partial grade points for that assignment. Always remember: homework assignments are a teaching aide, not mandatory military work (עבודות רס"ר). Please take all of this into consideration as you work on your assignments, and **good luck to all of you**!

2. How to share ideas while working on challenging assignments

The purpose of the homework assignments is for you to deal with class material by yourself. You should be able to do this without any external help, and working in pairs should help you tackle the more difficult challenges in every assignment. If you are unable to tackle a specific challenge by discussion with your partner, you may turn to external sources for help. There are many useful online resources with relevant programming tips. You should always treat information in external references with due caution and make sure that it fits the assignment's requirements. Asking other classmates for help is also okay, but this should be done verbally and **not by sharing code**. There are a few golden rules to help you know when you might be sharing code:

- If you put your code in a shared space, you're sharing code.
- If you send or receive a source file or copied text from a source file, you're sharing code.
- If you are writing your code while having someone else's code in front of you, you are sharing code.

When you share code, you increase the likelihood of similar bits of code appearing in two different solutions. We periodically scan for this, and when we find suspiciously similar bits of code in two different solutions, we take **disciplinary actions**, which can result in you being forced to withdraw from the course. All parties involved risk the same consequences (even if

you are the source), and this is clearly not worth the possible gain of a few grade points in one assignment!

3. Working on the course server vs. working offline and transferring files

Homework assignments are submitted on your home directory on the course server sysprog.runi.ac.il. You may log into the server via SSH / PuTTY to submit your solution. We highly recommend that you actually practice working directly on the remote server, because working "on the command line" is part of the skills you are expected to develop during the course. If you work directly on the server, submission should be quite straightforward (see **submission instructions** below). On the other hand, working locally has clear benefits, because it does not depend on an Internet connection and allows you to work with more powerful editors and integrated code development environments (IDEs). If you do prefer to work locally, then we strongly recommend that you install Linux on your machine as a parallel installation or using a virtual machine (VM). This will allow you to practice the Linux portion of the course and will help you avoid compiler-specific errors for your C code after you transfer files to the server. Regardless of where you choose to develop the solution to the assignment, you should always **check your submission**, make sure it has no fatal errors, and **submit it on the server**. *"It worked okay on my machine"* is not a valid excuse for a faulty submission!

4. What should your submission include?

Exercise solutions should be placed in the `~/exercises/ex?/` subdirectory in your home directory, where `?` is a single digit 1-6 indicating the assignment number.

Example: user `pablo.picasso` should submit his solution to Exercise 3 in the following directory path: `/home/pablo.picasso/exercises/ex3/`.

Each assignment will specify the files and directories that you need to prepare in your home directory. Before checking your submission, make sure that this directory contains all these files, plus an additional `PARTNER` file that includes the user id of your partner. Make sure to use the exact file names specified in the assignment! Any other files will be ignored by the submission script and will not be reviewed by the course graders. The `check_ex` script will help you make sure of that (see **checking submission** below).

Example: the directory structure required for Exercise 1 is:

- `exercises/`
  - `ex1/`
    - `my_commands.txt` `[text file]`
    - `PARTNER`

5. Submitting in pairs

You are required to submit all homework assignments in pairs. Working in pairs is a great way to learn and interact with your classmates. If you do not have a partner, please search for one via the Piazza website. Note that you may switch partners during the course.

Every submission directory `~/exercises/ex?/` must contain a file named `PARTNER` (not PARTNER.txt or anything else). This file should contain a single line with the user id of your partner. The partner should also place a `PARTNER` file in their `~/exercises/ex?/` directory with the submitting partner's user id. This will validate your partnership. If you eventually submit your work without a partner, then the file should contain a single line with your own user id.

The solution should be submitted by one of the students in the pair in their home directory. The `~/exercises/ex?/` directory of the non-submitting partner should contain a `PARTNER` file with the submitting partner's user id.

Example: Users with user IDs `pablo.picasso` and `claude.monet` work together on the solution for Exercise 4. They put the solution files in `pablo.picasso`'s home directory (`/home/pablo.picasso/exercises/ex4/`) together with a `PARTNER` file that contains a single line with the user id `claude.monet`. Additionally, they prepare another `PARTNER` file that contains a single line with the user id `pablo.picasso`, and put this file in `claude.monet`'s home directory (`/home/claude.monet/exercises/ex4/`).

6. Checking your submission

We prepared a script, named `check_ex` to provide you with useful feedback on your submission. The script receives the assignment id as an argument (e.g. `ex1` or `ex4`), and it checks the solution files you prepared in your `~/exercises/ex?/` directory. You may run this script **from any directory**, but before you run it you should make sure you prepared all required submission files, including the `PARTNER` file(s). The script tests the following three aspects of your submission:

1) **The `PARTNER` files.** The script makes sure that (1) you have a `PARTNER` file in your `~/exercises/ex?/` directory, (2) that this file contains a single line with a valid user id, (3) your partner has a valid `PARTNER` file in their `~/exercises/ex?/` directory and (4) that this file contains a single line with your user id. If any of these conditions is not met, the script will halt with an error message. Make sure that your `PARTNER` files contain only one line and that you correctly specified your partner's user id (lower case!). If you are submitting alone, the script will remind you that you need to ask the course staff for permission.
2) **The solution files.** The script will make sure that your `~/exercises/ex?/` directory contains all the files required in the assignment. It will also list other files you have in that

directory. These extra files will be ignored by the submission. If your `~/exercises/ex?/` directory is missing one of the submission files, the script will halt with an error message.

3) **Basic functionality of your solution.** Testing is an important part of software development and you should always allocate enough time for testing your solution. To help you with this, each assignment is accompanied with a series of automatic tests designed to help you validate your solution before submission. We provide these tests in the script `/share/ex_data/ex?/test_ex?`, and you may use them when you develop your solution. The `check_ex` script runs this test script and provides you with feedback. Some errors tested in this step are considered as **fatal errors**. You should avoid these fatal errors, as they will likely cost you many precious grade points (see below).

Checking your solution is vital for clean submission. Passing all three steps above without errors guarantees you a grade of 80 or up in the assignment. The 20 additional points are reserved for issues tested manually by the course graders. Fatal errors in submission are considered as any error in steps 1 & 2 above, or fatal testing errors in step 3. Fatal errors will cost you 40 points off your assignment grade (even for one small fatal error). Check your submission thoroughly and **avoid fatal errors!**

Examples:

1. User `pablo.picasso` is submitting ex1, but his non-submitting partner `claude.monet` did not provide a `PARTNER` file

```
==> check_ex ex1
+-------------------------------------------------------------------------------
| Checking ex1 for submission
+-------------------------------------------------------------------------------
| Step 1: checking partner file(s)
| Looking for partner id in file /home/pablo.picasso/exercises/ex1/PARTNER ...
| You are submitting with claude.monet
| Checking for partner validation in /home/claude.monet/exercises/ex1/PARTNER ...

Error: File /home/claude.monet/exercises/ex1/PARTNER with partner details cannot be
found!
-------------------------------------------------------------------------------
You need to correct errors in your submission format before it can be tested for
submission
Please correct and re-run 'check_ex ex1'
```

2. Solution is written in file with wrong name (`my_commands` and not `my_commands.txt`)

```
==> check_ex ex1
+-------------------------------------------------------------------------------
| Checking ex1 for submission
+-------------------------------------------------------------------------------
| Step 1: checking partner file(s)
| Looking for partner id in file /home/pablo.picasso/exercises/ex1/PARTNER ...
| You are submitting with claude.monet
| Checking for partner validation in /home/claude.monet/exercises/ex1/PARTNER ...
| Your partner claude.monet validates you as their partner
+-------------------------------------------------------------------------------
| Step 2: checking submission files and directories
|
```

```
| Required files:
|   file my_commands.txt :  not found!

Error: there are missing files or directories. Please prepare before submission!
+-------------------------------------------------------------------------------
You need to correct errors in your submission format before it can be tested for
submission
Please correct and re-run 'check_ex ex1'
```

3. Partial solution

```
==> check_ex ex1
+-------------------------------------------------------------------------------
| Checking ex1 for submission
+-------------------------------------------------------------------------------
| Step 1: checking partner file(s)
| Looking for partner id in file /home/pablo.picasso/exercises/ex1/PARTNER ...
| You are submitting with claude.monet
| Checking for partner validation in /home/claude.monet/exercises/ex1/PARTNER ...
| Your partner claude.monet validates you as their partner
+-------------------------------------------------------------------------------
| Step 2: checking submission files and directories
|
| Required files:
|   file my_commands.txt :  found!
| No missing files!
|
| Extra files:
|    Extra file: an_extra_file.txt
| Warning: Extra files will not be submitted!
+-------------------------------------------------------------------------------
| Step 3: testing your solution
|   Testing command file my_commands.txt
|   X - Your file is missing lines (expecting exactly 10 and your file has 0).
|   Fatal error:  missing commands 1 - 10.
|   Summary: 1 errors found, with fatal error - missing 9 commands.
+-------------------------------------------------------------------------------
You need to correct fatal errors in your solution before it can be submitted!
Please correct and re-run 'check_ex ex1'
```

4. Complete correct solution

```
==> check_ex ex1
+-------------------------------------------------------------------------------
| Checking ex1 for submission
+-------------------------------------------------------------------------------
| Step 1: checking partner file(s)
| Looking for partner id in file /home/pablo.picasso/exercises/ex1/PARTNER ...
| You are submitting with claude.monet
| Checking for partner validation in /home/claude.monet/exercises/ex1/PARTNER ...
| Your partner claude.monet validates you as their partner
+-------------------------------------------------------------------------------
| Step 2: checking submission files and directories
|
| Required files:
|   file my_commands.txt :  found!
| No missing files!
|
| Extra files:
|    Extra file: an_extra_file.txt
| Warning: Extra files will not be submitted!
+-------------------------------------------------------------------------------
| Step 3: testing your solution
|   Testing command file my_commands.txt
|   V - line 1 is okay.
|   V - line 2 is okay.
|   V - line 3 is okay.
```

```
|   V - line 4 is okay.
|   V - line 5 is okay.
|   V - line 6 is okay.
|   V - line 7 is okay.
|   V - line 8 is okay.
|   V - line 9 is okay.
|   V - line 10 is okay.
|   Your submission passed all tests, guaranteeing you grade 80-100
| If you are ready for submission, execute:
| ==> submit_ex ex1
|
+--------------------------------------------------------------------------------
```

7. Submitting your work

After you completed working on your solution and you checked it thoroughly to avoid fatal any errors (see above), you are ready to submit your work. All you need to do is run the script `submit_ex` (from any directory). The script receives the assignment id as an argument (e.g. `ex1` or `ex4`), and it copies all your submission files from your `~/exercises/ex?/` directory to a newly created `~/exercises/ex?/submit/` directory. The submission script also executes `check_ex` (see above) and it prints a report into an additional file in the `submit/` subdirectory. If your submission contains any errors, the script notifies you to correct these errors. The `submit/` subdirectory and all its contents are **write-protected** to avoid any changes after submission. Do not directly attempt to modify these files. If you wish to modify your submission before the deadline, modify the source files in your `~/exercises/ex?/` directory and re-submit them by running the `submit_ex` script with option `-o` (for overwrite). This will erase all previous files in the `~/exercises/ex?/submit/` directory and copy your modified files, generating a modified report file. Any modifications you apply after the submission deadline will not count, unless you received explicit permission to submit your work late (see **late submission** below)

Examples:

1. Trying to submit before checking solution.

```
==> submit_ex ex1
+--------------------------------------------------------------------------------
| Preparing to submit ex1
| Step 1: creating submission directory /home/pablo.picasso/exercises/ex1/submit
| Step 2: checking submission by running check_ex ex1 ...
Error: File /home/pablo.picasso/exercises/ex1/PARTNER with partner details cannot
be found!
--------------------------------------------------------------------------------
You need to correct errors in your submission format before it can be tested for
submission
Please correct and re-run 'check_ex ex1'
Your solution was not submitted due to fatal errors. See complete report in file
/home/pablo.picasso/exercises/ex1/submit/check_ex_ex1_2024_03_25_12_35.txt and fix
before re-submitting.
```

2. Initial submission with some errors.

```
==> submit_ex ex1
+--------------------------------------------------------------------------------
| Preparing to submit ex1
| Step 1: creating submission directory /home/pablo.picasso/exercises/ex1/submit
```

```
| Step 2: checking submission by running check_ex ex1 ...
|  Your submitted solution was found to have 3 errors when tested.
|  You may correct these errors and resubmit.
| Step 3: copying all files into submission directory ...
|   - my_commands.txt
|   - PARTNER
| Step 4: removing write permissions for submission directory
|
| Your solution to ex1 is now submitted (with 3 errors).
| For complete test report, see file /home/pablo.picasso/exercises/ex1/submit/
check_ex_ex1_2024_03_25_16_24.txt
| You may update submission by overwriting ('submit_ex ex1 -o')
|
+-------------------------------------------------------------------------------
```

3. Resubmitting modified solution (using the -o option)

```
==> submit_ex ex1 -o
+-------------------------------------------------------------------------------
| Preparing to submit ex1
| Step 1: overwriting existing submission directory /home/pablo.picasso/exercises/
ex1/submit
| Step 2: checking submission by running check_ex ex1 ...
|  Congratulations! Your submitted solution was found to have no errors.
| Step 3: copying all files into submission directory ...
|   - my_commands.txt
|   - PARTNER
| Step 4: removing write permissions for submission directory|
| Your solution to ex1 is now submitted (with 0 errors).
| For complete test report, see file /home/pablo.picasso/exercises/ex1/submit/
check_ex_ex1_2024_03_25_19_42.txt
| You may update submission by overwriting ('submit_ex ex1 -o')
|
+-------------------------------------------------------------------------------
```

8. Late submission

Your solution will be copied from your `~/exercises/ex?/submit/` directory a few minutes after at the deadline (which is typically at 9pm). Any modifications you make after the deadline will not be counted and may not be checked!

Requests for late submissions will be accepted in rare occasions and only if an e-mail request is made to one of the course instructors **48 hours or more** before the submission deadline. If you are not able to meet the submission deadline and are not granted a special extension, you may still submit your solution, and 10 points will be deducted from your grade for every day you are late. In that case, you should directly inform one of the instructors via e-mail or private message on Piazza so that we know to collect your solution from your home directory. Note that **the system does not automatically test for later submissions!**

9. Grader feedback and grades

Once your solutions are checked and graded by the course graders, we will copy a feedback file with your grades and detailed feedback to your `~/exercises/ex?/` directory. You should carefully review the feedback, and if you have any questions, you may post them as private notes to the course instructors on Piazza. If you wish to appeal your grade, you should post a private note to the instructors with the subject line: *"appeal grade of* `ex?`*"*. Label your

note with the *appeal* label, and clearly explain why you think your solution was wrongly marked. Your appeal should be short and to-the-point, and you should avoid questioning the general grading scheme (avoid arguing that *"this is a small mistake in my opinion"*). When you appeal a grade take into consideration that 10 points from one assignment are worth less than ½ a grade point from your final grade, and are likely not to make any difference.

10. Frequently asked questions

Should both partners submit the solution?

The actual submission, including all required files, is done in one partner's `~/exercises/ex?/` directory. Make sure to add an appropriately formatted `PARTNER` file in that directory and another `PARTNER` file in the second partner's `~/exercises/ex?/` directory. This second `PARTNER` file is only used for validation and is not a formal part of the submission. See **submission guidelines** for more information.

How do I find a homework partner?

The best way is to ask around class. You may also post a *"searching for HW partner"* note on Piazza. If all fails, contact one of the instructors and we will help you find a pair.

Can I submit my homework alone?

We really prefer that you work in pairs. We realize that this is not always easy, especially if you do not know your partner beforehand. This experience is a central part of your studies and it is important to develop efficient ways of working in small groups. If you could not find a partner or strongly prefer to work alone, then you should specify **your own user id** in the `PARTNER` file.

Can we submit in groups of 3-4?

No. Groups larger than 2 are strictly prohibited.

Can I switch homework partners?

Yes. You may switch partners as you choose. As long, of course, that this is done consensually. Just make sure that you submit the appropriate set of `PARTNER` files in each submission.

Can I use external references or consult other students when solving an assignment?

The homework assignments should make you deal with class material by yourself, and working in pairs should help you tackle the more difficult challenges in every assignment. If you are unable to tackle a specific challenge by discussion with your partner, you can turn to external sources, such as online resources and consulting other students. When you do this, you have to make sure that you are sharing ideas and not code. For more information on "dos and don'ts" (עשה ואל תעשה) when sharing ideas, see **this section**.

Can we modify our solution after submission?

Any time before the submission deadline you may modify your submission by modifying the source files in your `~/exercises/ex?/` directory and re-submitting them by running the `check_ex` script with option `-o` (for overwrite). This will erase all previous files in the `~/exercises/ex?/submit/` directory and copy your modified files, generating a modified report file. **Example:** `check_ex ex2 -o`

What is considered valid grounds for late submission?

In principle, we authorize late submissions for MILLUIM service and severe sickness. In both cases, you should notify an instructor as soon as possible, and provide the required documentation. Any request for late submission should be submitted **at least 48 hours** before the submission deadline. Last minute requests will not be considered. You always have the option to submit the assignment late with 10 points penalty per 24 hours. In this case, you should notify an instructor when you submit your solution, so that it is collected from your home directory.

How do I appeal my homework grade?

If you wish to appeal your grade, you should post a private note to the instructors with the subject line: *"appeal grade of* `ex?`*"*. Label your note with the *appeal* label, and clearly explain why you think your solution was wrongly marked. Your appeal should be short and to-the-point, and you should avoid questioning he general grading scheme (avoid arguing that *"this is a small mistake in my opinion"*). When you appeal a grade take into consideration that 10 points from one assignment are worth less than ½ a grade point from your final grade, and are likely not to make any difference.

Do all assignments count for the final grade?

Yes. We do not apply the n-1 rule in this course. The course has only six assignments and each one covers a different topic in the course. We therefore would like to encourage you to work on every assignment and submit your best possible solution. Partial submission will result in partial credit, and in case you skip one assignment entirely, you will lose less than 4 grade points from your final grade. It's not a lot of fun, but not the end of the world as well.