General guidelines for the programming exercises Basic Probability statistics and computing

Fall 2015, Master of Logic, University of Amsterdam

Philip Schulz & Christian Schaffner

last modified: September 6, 2015

1 Cooperation

Cooperation among students for both theory and programming exercises is strongly encouraged. However, after this discussion phase, every student writes down and submits his/her own individual solution.

2 How grading will be done

You will receive your grades through peer assessment. Each student will receive reviews from three other students and the points for an assignment will be the average of the points given by the reviewers. Students can receive at most 10 points per assignment. In addition, each student will receive an assessment of their peer grading efforts from the instructors once during the course. To this end, a couple of students will be chosen randomly each week whose reviews will be double checked by the instructors.

3 Why peer grading?

The idea of peer grading is very simple. By assessing other peoples work you learn to a) read and understand other people's code b) give feedback in a constructive manner c) get inspiration out of other people's work. The third point may be the most important to your personal development as a programmer. Seeing other people's code can be of tremendous help especially when you have just started programming. You basically get to pick and choose programming patterns from different authors that you can incorporate into your own coding style or avoid if you find that they do not work for you.

4 Rules of peer grading

To make peer grading a pleasant and educating experience for all of us, here are some general rules on what you should and should not include in your reviews.

- NEVER include any information in your review that could be used to identify you as the reviewer.
- Give verbal feedback! Just deducting points won't be helpful to the person you are reviewing. Try to be precise about why you deduct points and explain what the person could have done better.
- Even if you give full points, try to suggest improvements. For example, could the person have made their code shorter or more readable? If so, tell them!
- Try to be constructive in your criticism. Instead of writing "This is just bad" write something like "This part of the code could be made better by doing X".
- Be polite! Do not attack the person who wrote the code but solely comment on the code itself. Do not use any inappropriate language!

5 Submission

Each of you will receive their own url to which to upload their programming assignments. This is your personal url and it should be kept private! There will be three folders: submissions, reviewing and feedback. The submissions folder is where you place your completed assignments. The reviewing folder will each week contain the assignments that you have to review. Finally, the feedback folder will contain the reviews that others have written for you.

For each assignment you will need to submit a file or folder (package). Your submission should be named

assignmentX_yourStudentNumber(.py)

where X is the number of the assignment and yourStudentNumber speaks for itself. The .py extension is required if you are submitting a single Python file. If you submit a package, no extension should be attached. Luckily this is a point that you will not need to worry about, as PyDev takes care of this. Each Python file that you submit (this includes all files in your package) should include the line

# author: yourStudentNumber

at the beginning. Please make sure to not include any information (other than your student number) that identifies you as the author.

6 Deadlines

The deadline for the programming assignment of week i is Monday, 9 a.m. of week i+1. The deadline for the reviews of the assignment from week i is Tuesday, 9 a.m. of week i+2. We will be very strict about those deadlines. Your submission time is the time stamp in the online folder. Notice that uploading may take some time, so make sure to upload your assignments and reviews well before the deadline. Also notice that viewing (and of course modifying) your files may change the submission time stamp. Make sure to not touch them anymore after the deadlines!

Of course we understand that there may always be extenuating circumstances. Just make sure to let us know well in advance. Crucially, you need to inform us **BEFORE** the deadline has passed.

7 Grading guidelines

For each assignment, their will be specific grading guidelines which explicitly tell you what to give points for. Important: if these guidelines are at any point unclear and you are unsure about how to give points, we have failed in specifying the guidelines correctly In that case you should immediately contact Philip and ask for clarification.

When you grade, write the points and verbal feedback as comments directly into the code. For better readability, please begin each of your comments with

# REVIEWER:

At the very end of the code file you should sum up the total number of points.

# TOTAL POINTS: X/10

where X is the sum of the points you have awarded.

To finish, here is a list of things that will automatically lead to point deduction:

- If your code cannot be executed in Eclipse or the command line, you automatically receive 0 points for that assignment!!!! (to be done by the reviewer)

- If you do not put a docstring into ALL your functions, -2 points (to be done by the reviewer)

- If your submission does not comply with the format specified in Section 5, -1 point (to be done by the reviewer)

- If your submission is late by one day, -2 points (to be done by the instructors)
- If your submission is more than one day late you receive 0 points!!!! (to be done by instructors)
- For each review that is late, -1 point (to be done by the instructors)