UNICAM
Università di Camerino
1336

G-Node German Neuroinformatics Node www.g-node.org

Advanced Scientific Programming in Python

a Summer School by the G-Node and the University of Camerino

3—8 September, 2018. Camerino, Italy

Evaluation Survey Results

Method

The survey has been administered with a web interface created with the LimeSurvey software available at: http://www.limesurvey.org
All answers have been submitted by 6 October, 2018.
No answer was mandatory.
The free-text answers have not been edited and are presented in their original form, including typos.

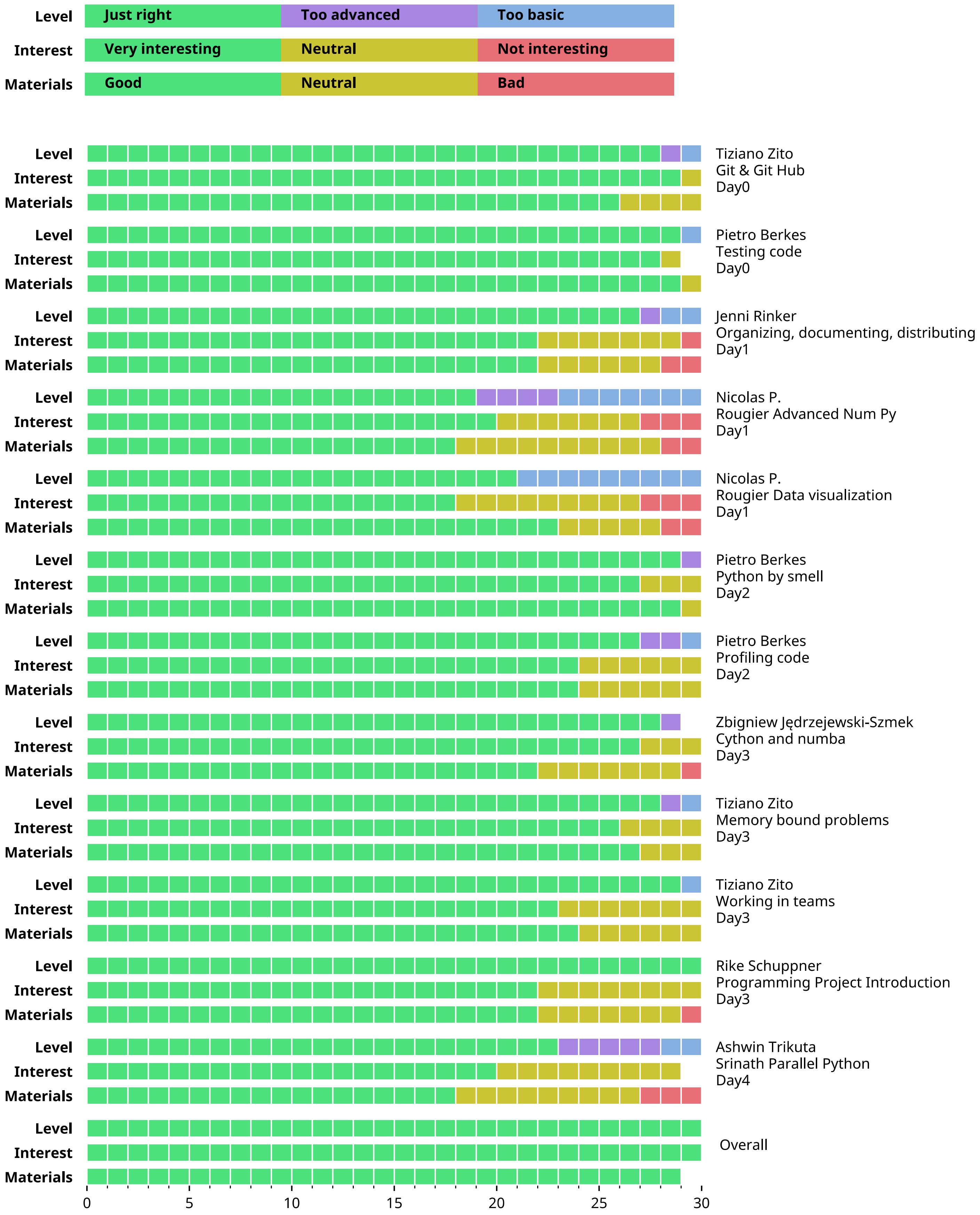
Attendants and Applicants Statistics

	Attendants		Applicants	
	30	16%	188	
Different nationalities	17		43	
Countries of affiliation	13		29	
Gender: other	0	0%	0	0%
Gender: female	13	43%	66	35%
Gender: male	17	57%	122	65%
Already applied	13	43%	28	15%
Bachelor Student	1	3%	14	7%
Master Student	4	13%	38	20%
PhD Students	19	63%	81	43%
Post-Docs	4	13%	24	13%
Professor	2	7%	3	2%
Technician	0	0%	1	1%
Employee	0	0%	14	7%
Others	0	0%	13	7%
Completed surveys	30	100%		

More stats about attendants are available at: https://python.g-node.org/python-summerschool-2017/students

Lectures & Exercises

Q: Grade the level of the lectures
Q: Grade how interesting were the lectures
Q: Grade the quality of the presentation style and/or of the teaching material provided by the lecturer, e.g. the clarity slides, the exercises and the solutions, etc.



Q: Are some of the topics presented in the lectures not relevant for a programming scientist?

1. Advanced numpy itself seems very relevant, just not what was presented during this lecture (the strides). That seemed too case specific. maybe more about common 'mistakes' where numpy could be used to do it more efficiently?
2. The Cython/numba and memory-bound problem lessons were very interesting but mostly useful to scientists who work with very large datasets.
3. All are relevant.

Q: Are there further topics relevant to the programming scientist that could have been presented, given that the total time is limited. Please also mention which topics should be replaced by the new ones.

1. I'd give more time to Nicolas to talk about numerics on the cost of presentation.

   In my view one hour for presentation would be enough; his summary on available plotting frameworks was quite useful (as it is hard to get a broad perspective), but I would remove from the program the one hour for Matplotlib basics. I think everyone can go easily through that on their own.

   In gained time I would add some info about SymPy - just so that people know that it exists and when to use it and maybe some short overview on SciPy. In general, I think it is very valuable to get broad overviews, e.g. I never heard about ATLAS/BLAS before and now as I know what they are and when to use them, I expect it would be not very hard to follow some manuals and simply do it. In the same spirit, I think it would be useful to give an overview of even more python scientific tools, e.g. not necessarily spend two hours on pandas, but just say that it exists and what it is. If anyone ever needs it, they will know where to go.

   I'd add some short info about licensing to Jenni's lecture.

   I'd remove the part about classes from Pietro's lecture, I think every engineer goes through classes on object-oriented programming. There was too little time to talk about operators, I'd rather listen about that instead. Finally, I'd add 10 minutes to discuss how Python works in terms of value/reference passing, I remember I was surprised by some Python behaviour in the beginning and am quite sure for many people it would also be new.
2. Best/good practices in debugging code
3. I think given the time constraints, the topic selection is appropriate.

   Maybe Python Notebooks could be introduced, but that's just because I am not so familiar with their use.
4. Just to start out: Short recommendation for which editors are good (for debugging, looking at variables, stepping through code) as well as how the different python versions differ (basic python installation vs anaconda, using package managers; installing your packages the right way and in the right places and keeping the python installation clean)
5. More data vizualization would be great, maybe by cutting a bit of the parallelization.
6. More time could be dedicated to how to organise large coding projects and how to do proper documentation.

   Maybe the memory bound problems (which seems to be taken care of automatically by numpy/numba) could be replaced?
7. Personally, I'd rather have learned about decorators than classes. And I would have liked hear something about efficient debugging (or debugging tools or something). Just not sure which should be replaced. Maybe the parallel python? I thought it was very interesting, just that it all the way down in the list of things you should do to improve your code
8. sk-learn tensorflow or pytorch: 10% (1)
9. For instance, best practice for working on servers, this topic can be combined with parallel Python.

   How to organize data in a decent way might also be beneficial, for example, which data types are more appropriate to manipulate and for later stats. Perhaps one could introduce an "Advanced Panda" session combined with "Advanced NumPy".

   Lastly, a more advanced debugging session might also be helpful.
10. Perhaps spend less time on speed and more time on advanced numpy, which felt a bit short and rushed.

Q: Do you think that pair-programming during the exercises was useful?

Yes, I have learned from my partner / I have helped my partner	93% (28)
No, it was a waste of time for both me and my partner	3% (1)
Neutral. It was OK, but I could have worked by myself as well.	3% (1)
Other	0% (0)

Q: What do you think of the balance between lectures and exercises? When answering, please keep in mind that the overall time is limited ;-)

Lectures were too long, there should be more time for exercises	10% (3)
Lectures were too short, there should be more time for lectures	0%
The time dedicated to lectures and exercises was well balanced	87% (26)
Other	3% (1)

Other: Overall the balance was excellent. Some lectures could have benefitted from more exercises - such as the advanced Numpy lecture. I think most of us needed more time to understand and implement code using strides.

Q: Any further comments about the lectures and exercises?

1. ASPP changed the way I think about and write my code. In the first week back I already implemented two principles I learned at the course, so it was very relevant and extremely helpful.
2. I have the impression that the exercise sessions that were more effective were those that were using Jupyter notebooks for the exercise material. With the notebooks it was easy to see what exactly was the task to be solved and check the theory or the examples to get an idea of how to solve the exercises. In other lectures, the exercise to be solved sometimes would not be clearly stated and this lead to confusion among the students, sometimes the lecturer had been typing at the keyboard and the information needed to solve the exercise was not available.
3. The lectures were excellent and full of relevant and interesting information. The exercises were the right level and were fun to work on, especially together with a partner.
4. On some occasions, the exercises were a bit too simple and boilerplate. And on other occasions we started but were interrupted too quickly and too often, not having the time to tackle the problem.
5. The way the lectures and exercises were structured was very nice, I had no trouble staying focused.
6. The material was both interesting and high quality, and the teachers were super engaged. It was almost sad to reach the end of the daily lectures, even after 9 hours of class.

   I believe using the provided computers instead of each student's private laptop helped a lot to focus by preventing distractions, and working in pairs was great. On one side it involved a lot of learning and teaching between the pairs, and as a bonus it increased the individual commitment to the class.
7. I found the course to be targeted to precisely my level of familiarity with python and needs for further research-specific training.
8. I would prefer working with my own laptop.
9. In my opinion the course was well thought and well organized, every topic was presented in an interesting way and the ratio between theory and exercise was optimal. Well done!
10. Great job!
11. It was a great summer school! I learned a lot! Many thanks to the organizers!
12. It is hard to find the materials, each class has a different place. Even now, after the lectures, there is no mainpage where we can find them.
13. I am so glad to having had the opportunity to come to Camerino and taking part in this amazing summer school. It was very well organized, intensive (but in a good way) and incredibly helpful, than you very much
14. Overall a great course with excellent material! Helpful to my research and a fun experience.
15. The summer school was amazing! I've learned so many useful things in only one week and they are already improving my work. Thank you so much for this opportunity!
16. I felt that overall the lectures and exercises were excellent. The only way that I could see improvement is by extending the time of the school, sometimes there was too little time to perform the exercises.
17. Let me start with the complement: thanks for all the effort and love you guys have put into the summer

school and all the lectures and exercises!

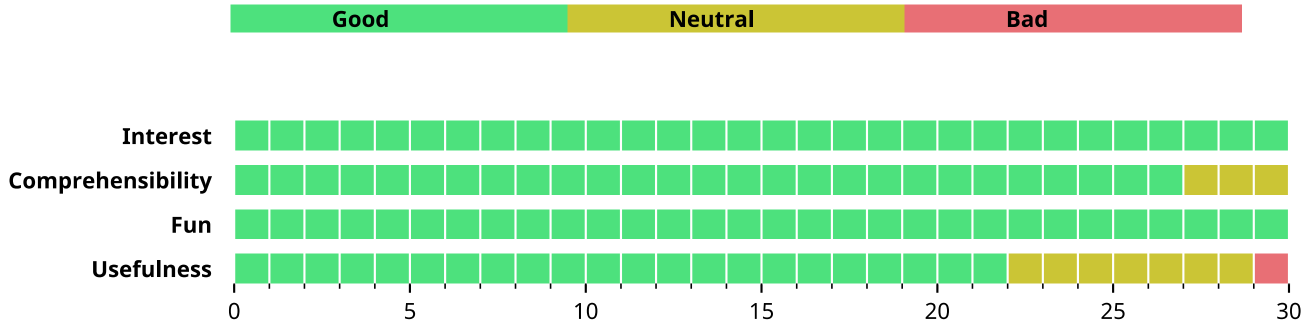
I would assume if there is a better attention catching strategy (other than whistling, which is good, but IMO using it only 1 time or two is efficient). The idea is that, when the lecturer wants to talk about something, usually some tips relevant to the exercises or to Python in general, if students are still discussing, it is unfair to those who want to pay attention to the lectures. In my opinion, listening to the lecturers are way more important than solving an exercise.

18. The program was very interesting, engaging and clear. The team programming excersizes made it more social and people could help each other out, which worked perfectly!
19. I found the lectures and exercises overall very interesting, well prepared and presented and really useful to improve my programming skills.
20. The fact that almost everything was published on Github for further reference - very good idea. The length of lectures and breaks was most of the time very well balanced. Big plus also for the natural light in class and for the post-it comunication when the pair was ready. Many members of faculty helping during the exercises made us not afraid of asking questions. In general I feel that I have learned a lot and every lecture contained something new for me.

Programming Project

Q: Evaluate the programming project.

Interest: How interesting was the programming project?
Comprehensibility: How clear and comprehensible was the code and the available documentation? Was it easy to work on the programming project
Fun: Was it fun to work on the programming project?
Usefulness: Was it useful to work on the programming project? Do you think you may re-use what you learned?



Q: Do you think the team-programming experience is relevant to your work as a programming scientist?

Yes: 93% (28)
No: 7% (2)

Q: Do you think that the project should be about a real-world scientific problem instead of a video game?

Yes: 7% (2)
No: 93% (28)

Q: Any further comments about the programming project?

1. Mostly, I found the project helpful in practicing my skills with github and testing/profiling code. However, I think it might be helpful to emphasize how groups were selected (presumably to create teams balanced in regards to experience and comfort with the material) and why pair programming was used. I found that some more advanced participants started to leave their partners out of the process for the sake of coming up with solutions more speedily. Learning rather than winning should be emphasized (perhaps providing more time for the final project would help with this).
2. The programming project was extremely fun. Also, having a repository where other people contribute to and learning how to fix the conflicts that arise from this under time pressure is useful.
3. Again, the only improvement I could see would be to extend the school to further implement the learned subjects to the participant's own personal scientific projects. For example, at the end one could make a project in order to create a library of functions/classes for a personal project together with tests, setup files and be able to package it, publish it online and distribute it.
4. it was super fun.

   it would be nice if the source codes of other groups/tutors are open.

   also id like to make another new bot in my free time and publish it somewhere.
5. Great fun! And was a good summary exercise for some things we have learned in the lectures before.
6. I think it was fun to work on the project and have the little tournament in the end. In my opinion, it is good to use the game instead of a scientific problem, because for the game, everyone was on the same knowledge level and after the week we are all going back to a lot scientific problems any way, so it"s nice to see something else in between.
7. I think the competitive aspect of the programming project was a lot of fun but definitely discouraged good programming practice.

   A more useful type of competition might be to a) test if there are bugs in the code then b) test the % coverage of the code with tests and c) take random documentation strings and see if you can use this to figure out what the functions are doing!
8. My team was very well balanced, thanks to the entertaining project we all got very involved. It was possible to share the tasks, that was also very good. The only drawback was sometimes the level of sound in the room - too many groups, too little space
9. We had so much fun:-)

10. The output of the programming project is focused on the efficacy of the team's bots. Perhaps there would also be a way to incorporate a component that requires integration of CI testing and strategy documentation to help participants practice the skills taught earlier in the course.
11. The programming project was a fun way to get us to use what we learned in the course without the usual stress of doing something related to our work. The tournament in the end was an absolute highlight and I would not change a thing about it.
12. It was very interesting. The teamwork was a great opportunity to put in practice most of the content of the lectures. The API provided by the faculty was very easy to understand and the documentation was clear. This let us focus on programming the AI of the game.

    Overall a great experience and a lot of fun. Also the final tournament was very exciting and had quite an epic final!!
13. I said "no" for using a real-world game. Because students are from different fields and they have distinct expertise. If a real-world project is used, it will definitely be much more beneficial to some, but more likely be boring to others.

    Using a video game instead does not depend on any scientific knowledge, and interesting enough for students to exercise the best practices of Git/Github, testing, and so on.
14. The project and the group work was a new environment to work in for me. The group work did teach me a lot but it was hard to estimate what would be possible in the given time, and good programming practices suffered a bit toward the end, when people were panicking about the deadline. I would have liked to apply more of the learned material to the project.
15. The programming project was one of the highlights of the school for me. It was a great and fun way to apply all the things we learned. I encourage 100% to keep this up :)
16. Great fun, perfect to learn to program in a team.
17. I think using a game for the project is great because it adds levity and fun to the situation. People tend to put pressure on themselves to perform as it is, and I think making it into a game helps to remind people that it is a learning exercise.
18. I think the programming project was a great idea, and the fact that it was a game and thus neutral avoided any issues that would come from a select topic. Would be nice to have more time, but it was limited so it is just right.

The School in General

Q: How do you overall evaluate the school?

Good: 100% (30)
Neutral: 0% (0)
Bad: 0% (0)

Q: How do you evaluate the general level of the school? Was it too advanced/too basic with respect to your expectations?

Too advanced: 3% (1)
Just Right: 93% (28)
Too basic: 3% (1)

Q: How do you evaluate the general level of the school? Was it too advanced/too basic with respect to what was advertised in the announcement?

Too advanced: 0% (0)
Just Right: 97% (29)
Too basic: 3% (1)

Q: Did you learn more from attending the school than you would have learned from reading books and online tutorials alone?

Yes: 100% (30)
No: 0% (0)

Q: How do you evaluate social interactions and social activities at the school?

Good: 93% (28)
Neutral: 7% (2)
Bad: 0% (0)

Q: Would you recommend this course to other students and colleagues?

Yes: 100% (30)
No: 0% (0)

Q: How did you hear about the school?

Google Search: 8
Professor/Tutor/Supervisor: 1
Colleague/Friend: 15
Mailing list: 12
Other: 2

Q: Any further comments or suggestions?

1. This was really an incredible experience and I've already recommended the school to several colleagues and friends! I also thought having a mixture of backgrounds and skill levels was an advantage of the program, and I did not observe the more advanced students getting annoyed by working in pairs/groups with less experienced students. I also thought a strength of the school was that the students work in such different fields, yet we all converge on needing software to get the job done. Is the diversity of students' backgrounds included in the admissions algorithm, or did this just happen naturally?

   Suggestion: It would be great to have a short talk about professional options outside of academia. Some of the lecturers had made the academia-industry transition, and quite a few students who are doing PhDs are also considering that transition. I think even a short talk on this would be relevant and interesting for many of the students.
2. The balance between too basic / too advanced was perfect (at least for me).
3. The summer school exceeded my expectations. Thanks to the spirit of the organisers and the high level of the content, I would rate this summer school as one of the best ones I have ever attended. The combination of people was also excellent and made the experience even richer and more enjoyable.
4. I think the school was great and I am very glad that I have attended.
5. I just wanted to say again how much I liked it, and how excellent I found it. Even when the group situation in the programming project was far from optimal, I found that to be a super instructive experience. And faculty

is great, students get so much of attention and feedback. Please please keep on doing this!

6. Keep up the good work!
7. Thank you so much, it was amazing and very helpful. Keep it up, such that many future students may benefit from this extraordinary schoold
8. Thanks a lot to all organizers, good job! I felt that I don't need to worry about anything during the school (i.e. Camerino - Rome travel, length of the lunch break, prebooked accomodation)
9. I love all aspect of this school! I will also experiment "working in pairs" style in my own teaching, which I think, should become standard for summer schools/workshops like this.

   I wish all my best for the future of ASPP, and I will certainly recommend it to my colleagues, and friends alike!

   Grazie!
10. The school was an amazing experience! I have learnt a lot about scientific Python programming in an extremely enjoyable environment and had the opportunity to meet awesome people (faculty and students) which are doing really interesting work in very different areas. I feel during this week I have really improved my programming skills and I also have a lot of new material and information to explore further.

    All in all I think this summer school is perfect for scientists who want to become better programmers, I am very grateful of having had the opportunity to attend and I highly recommend it to anyone interested in the topic.
11. Awesome job organizing this course! :)
12. Keep the workshop running, it's amazing! It's extremely useful and valuable to the whole scientific community.
13. The organisers did a wonderful job making everybody feel comfortable and created a great learning environment alongside a fun and social atmosphere.
14. - The schedule could have been clearer.

    - info about the dormitory: what is available and what not. such as cooking tools and laundry.