

## Advanced Scientific Programming in Python

a Summer School by the G-Node and the Institute of Experimental and Applied Physics, Christian-Albrechts-Universität zu Kiel

September 2–7, 2012. Kiel, Germany

### 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 October 12, 2012.

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	21%	141	
Different nationalities	18		43	
States of affiliation	12		25	
Female	6	20%	21	15%
Already applied	18	60%	30	21%
Bachelor Student	2	7%	7	5%
Master Student	3	10%	17	12%
PhD Students	19	63%	73	52%
Post-Docs	4	13%	25	18%
Technician	1	3%	2	1%
Employee	1	3%	8	6%
Others	0	0%	9	6%
Completed surveys	25	83%		

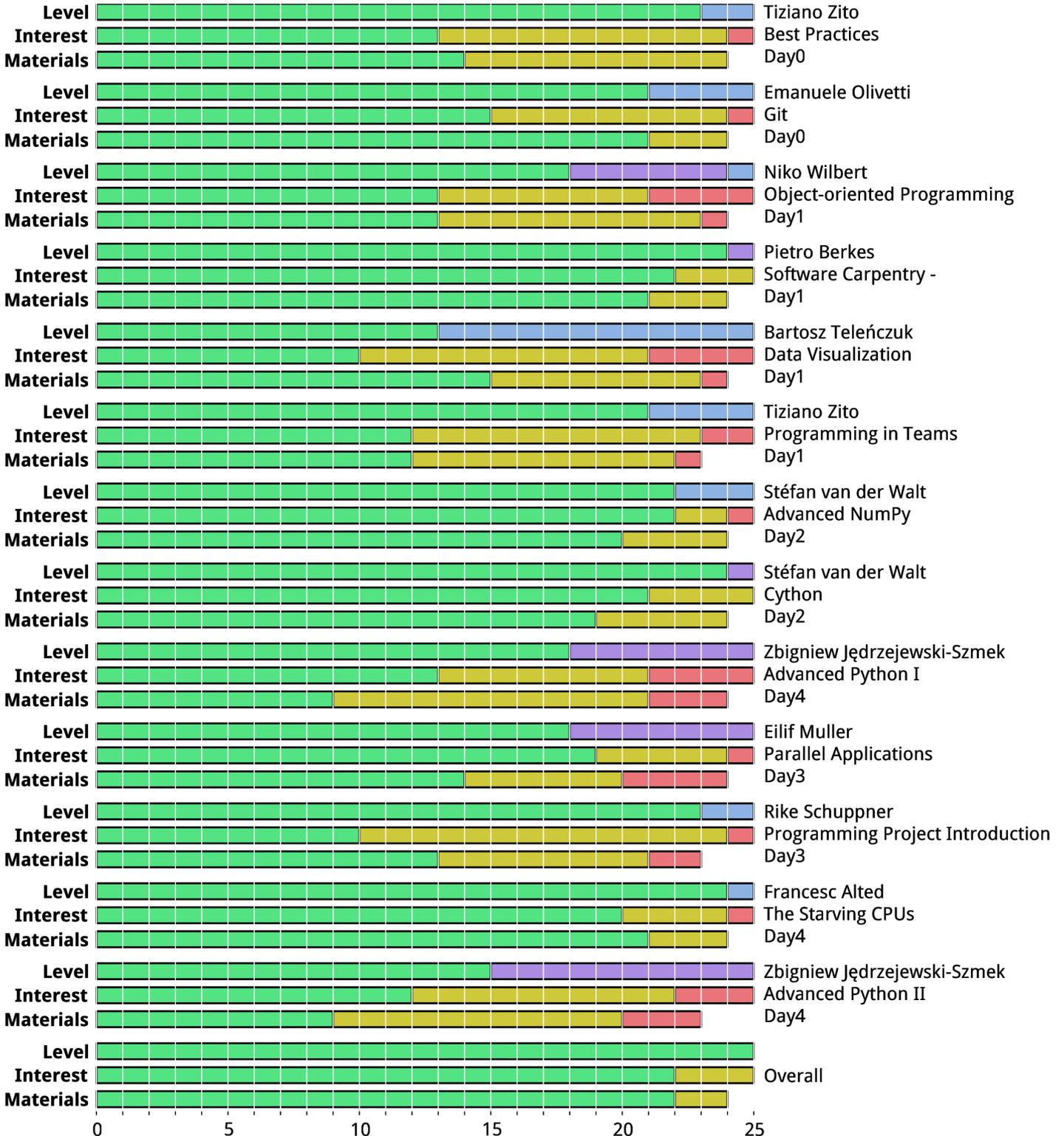
# Lectures & Exercises

**Q: Grade the level of the lectures**

**Q: Grade how interesting were the lectures**

**Q: Grade the quality of the teaching material provided by the lecturer, e.g. the clarity of the slides, references given, exercises etc.**

Level	Just right	Too advanced	Too basic
Interest	Very interesting	Neutral	Not interesting
Materials	Good	Neutral	Bad



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

1. The advanced programming techniques and the object oriented programming lectures are interesting for scientists. Nevertheless, some examples taken from science would have help to understand the notions and also provide reusable code snippets for the scientific work.
2. Some obscure data structures, like the two-headed list, are of little use in the general case, and they yield a relatively small performance boost even for their optimal use case.
3. I think all topics were relevant but not for all students. For example for a beginner object oriented programming was too advanced.

**Q: Are there further topics relevant to the programming scientist that could have been presented, given that the total time is limited**

1. Although that is arguably difficult to fulfill due to different platforms (Windows, Linux, MacOSX): how to set up a working, maintainable and extendable python environment?
2. No, selection of topics was perfect
3. More on how to setup some systems: like how to set up a parallel or distributed system and interface with python
4. Databases, Django
5. It could be useful to have a class on efficient algorithms for common analyses (eg. pca, ica, clustering etc.) or, since efficient methods already exist, a discussion of which methods are best, and why
6. I think it was a really thorough overview.

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

Yes, I have learned from my partner / I have helped my partner	72% (18)
No, it was a waste of time for both me and my partner	4% (1)
Neutral. It was OK, but I could have worked by myself as well.	8% (2)
Other	16% (4)

Other:

1. It may be useful if both partners know each others programming skills, but i prefer to work alone or to program in distributed teams online over git repository.
2. Highly depends how me and the partner are tuned at the same frequency
3. It was helpful, but I would have enjoyed it more if everyone had been willing to support this (and had left their laptop in the bag!)
4. All three apply, depending strongly on partner.

**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	8% (2)
Lectures were too short, there should be more time for lectures	0%
The time dedicated to lectures and exercises was well balanced	71% (17)
Other	21% (5)

Other:

1. Many times speakers had to hurry at the end and usually I had no time to do all excersises, but it's fine, time was limited and one can finish excersises on his/her own later
2. It should be more lectures, maybe exercises should be provided only for some lectures during school.
3. Some lectures could be shorter. The exercises covered the whole lectures but often we couldn't solve all of them.
4. 1:1 time for lecture and exercises would have been great - as it was planned - if the lectures would not take longer than planned...

5. Very well balanced, if kept better to schedule

**Q: Any further comments about the lectures and exercises?**

1. The main problem I think is the time distance between theory and exercises. While you are listening, things appear feasible, but after 90 min, they add up a lot, and many things are forgotten. I think it would be good to have small exercises during the lecture, to apply no more than the very basics in a quick problem. This way, you can make sure you actually understand it (to some extent) and remember it better.
2. The lectures usually started with the introductory easy stuff, but many of them suddenly bursted into really advanced issues with almost no transition. This way, one gets relaxed at the beginning (if more or less knows this), but before you realise things are getting hard. I suppose this is something unavoidable when you have people with very different backgrounds.
3. It may have been better to put the tutor consultation time in the morning, because in the afternoon all (students of course, because the tutors are amazing;) are a bit tired. Again thank you very, very much for this high quality summer school.
4. Maybe next year there should be planned key signing party. :) More advanced git usage should be explained especially if all of our tournament players are basically in one file that generates conflicts in merge. If you are planning to keep the idea of pair programming there should be separate system account for each student. Maybe you can publish software requirements for student's notebooks if they prefer to use their well-known set up. Maybe some kind of package or just the school dependencies list. :> And the last but not least you should provide an archive to all of teams players. They all should be GPL licensed, right? You can do it for this year tournament too. And of course Tiziano should give up his gedit idea to emacs or vi at least. Maybe it's good idea of having the lecture about emacs with exercises of course. In the end it is a part of good programming practice, right? I enjoyed it a lot, it was pleasure to be there!
5. how about a) making all lectures in a single block during the morning (with shorter lectures, see last comment), then b) a short recap of the presented topics after lunch, then c) start with exercises with one break plus progress report (this would mean of course that partners are switched only per day)
6. Big thank you!
7. I think it was important to have the slides of the lectures and the solution of exercises. I could go over the slides and practice and solve the exercises again. It is very helpful to have better commented slides.
8. The evaluation survey (at least for lectures and exercises) should take place during the school, for example every second day (better comparison between lecturers). After a week or so the answers now are not so accurate ...
9. Very good: Often lots of exercises, possible to pick the interesting / suitable to my needs. If Zbigniew would not have tied up his red wire sometimes, would have been even better understandable (but was well anyhow). If "decorators" would have come before "oop", my understanding of "oop" would have improved. Over all: a very good choice of experts, all knew what they talked about, it was a pleasure to listen!
10. The time balance between lectures and exercises was good, and the nature of the exercises allowed us to do as little/as much as we thought was useful. I would keep the overall format as it is right now.
11. I found that the numpy lectures were to basic given that one was already urged to go through the introductory material which was basically about numpy.
12. Most tutors needed longer time for their lectures than what was planned. When they realized they were over time, they just speeded up - I didn't like this. Better would be to put less content in the lectures (and only give references to other potentially useful functions, packages etc...), ie., more "time-discipline" from the tutors.

# Programming Project

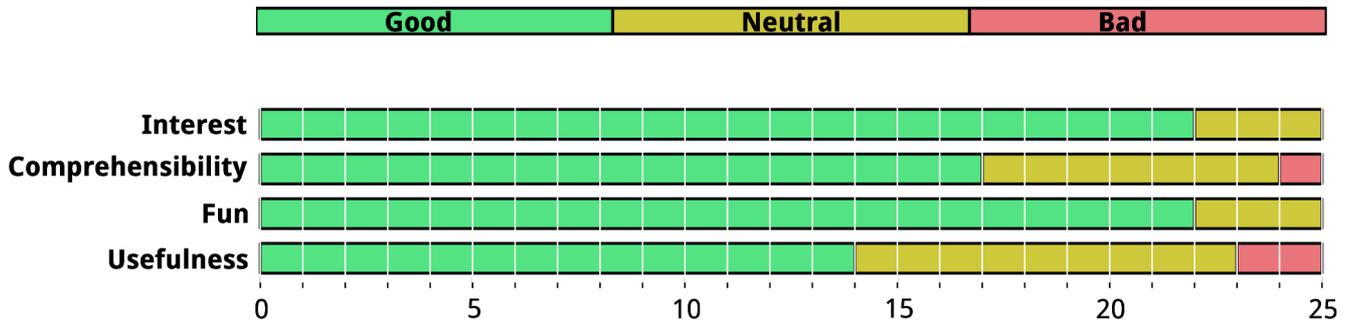
## Q: Evaluate the programming project.

**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: 76% (19)

No: 24% (6)

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

Yes: 8% (2)

No: 88% (22)

## Q: Any further comments about the programming project?

1. If I understood it correctly, the teams were put together randomly. As a result, the distribution of python experts and beginners in the teams was imbalanced. For example, me as "advanced beginner" did not feel very useful in a group full of experts (eg., until I managed to get the repository, my group mates made first commits etc.).
2. I really enjoyed the project, and in fact I've been playing around with the game since the course, designing new players. It might be good to have a project that requires you to use all the methods covered in the course, but obviously this would be difficult. The most important thing is that the project is interesting, and i think this one was.
3. You should recompile the documentation, and you should explore the end-game exception issue that NoTimeToCheckIt found. It was fun to participate in the game! Maybe there is too much logic already built in, so it's easy to win writing code alone. It would be much more difficult to win and the tournament would be much more fun if everybody were not using the same Astar and BFSsearch functions to move their bots around.
4. It was great fun to work in teams and combine thinking about strategies/tactics with coding the actual bots. It's much better to work on a project like this because working on specific scientific projects would always mean picking a topic that may be relevant for some participants but difficult to comprehend/relate to for others. Having the tournament was a really great way to end the summer school and the team experience definitely made the school much more useful than if it had simply been lectures all the way.
5. For me (Python beginner) it was hard to dive in the complex principle. From saturday on, it was ok, but too less time left. While reading the docs again afterwards (with more time) it became better clear and I suddenly found almost all answers to questions which took hours before. Suggestion: Maybe explain the docs in the introduction, where to find what; the sources are the most helpfull imho, and well documented. In real life it would be ok, but then I usually have enough time... Since our team was more of a bunch of special characters, I only learned: choose your partners carefully in real-live ;-). Overall it \*was\* fun :-)
6. This project is a good idea because it is an open problem, but the basics are easy to understand no matter which is your scientific background. Our team blatantly lost because some strategies were tuned against very poor teams. If we had any good team to test with, we may have tuned better our strategical parameters and had a better chance. This team could be distributed as a .pyc file, so we don't see what is underneath.
7. Team members bit too unequal, to achieve the best result the team had to let the more experienced programmers do major tasks.

## The School in General

### ***Q: How do you overall evaluate the school?***

Good: 100% (25)  
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: 0% (0)  
Just Right: 88% (22)  
Too basic: 12% (3)

### ***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: 84% (21)  
Too basic: 16% (4)

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

Yes: 96% (24)  
No: 4% (1)

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

Good: 92% (23)  
Neutral: 8% (2)  
Bad: 0% (0)

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

Yes: 100% (24)  
No: 0% (0)

### ***Q: How did you hear about the school?***

Google Search: 2  
Professor/Tutor/Supervisor: 6  
Colleague/Friend: 11  
Website/Mailing list: 8  
of which:  
    connectionists/comp-neuro/ML-news: 3  
    python: 1  
    debian: 1  
    bccn/g-node: 2  
    other: 1

### ***Q: There might not be further editions of the school unless we find a way to make it a self-supporting event. Would you have attended the school even if a fee were introduced to cover the running costs?***

Yes: 88% (22)  
No: 12% (3)

### ***Q: If yes, do you think a fee of about 150 € would be appropriate?***

OK: 71% (15)  
Too high: 19% (4)  
May be higher!: 10% (2)

**Q: Any further comments or suggestions?**

1. Some lectures suffered little bit from too long introduction part leaving not enough time for the main message and programming patterns and the following excersises.. In tight schedule I think it would be necessary to jump straight into the core sooner.
2. The content of the course is well worth much more than 150€, but charging a fee could change the course significantly. It just would not be the same.
3. The summer school was more useful to me than a lot of conferences I have been too, and they can often charge quite high fees, so 150 Euros seems quite low/reasonable. I assume most people would be able to claim a fee back from their department, so maybe a lower fee for students and higher fee for none students would be appropriate, as students often have a lower travel budget.
4. The school was great! I really learned a LOT and even enjoyed it!
5. Just make future editions even if fee will be introduced to cover the running costs.
6. It was a really good experience and I'm very happy to have attended. I learnt a wealth of things, from important concepts to many many small tricks and more efficient ways of doing day-to-day tasks. The direct contact with the lecturers was great, the interaction with the other students and the social events were really good too. Overall a great experience that I will be happy to recommend to others!
7. I miss checkboxes "very good" and "excellent"!
8. Thank you very much for making this summer school happen!