5TH HEP C++ COURSE

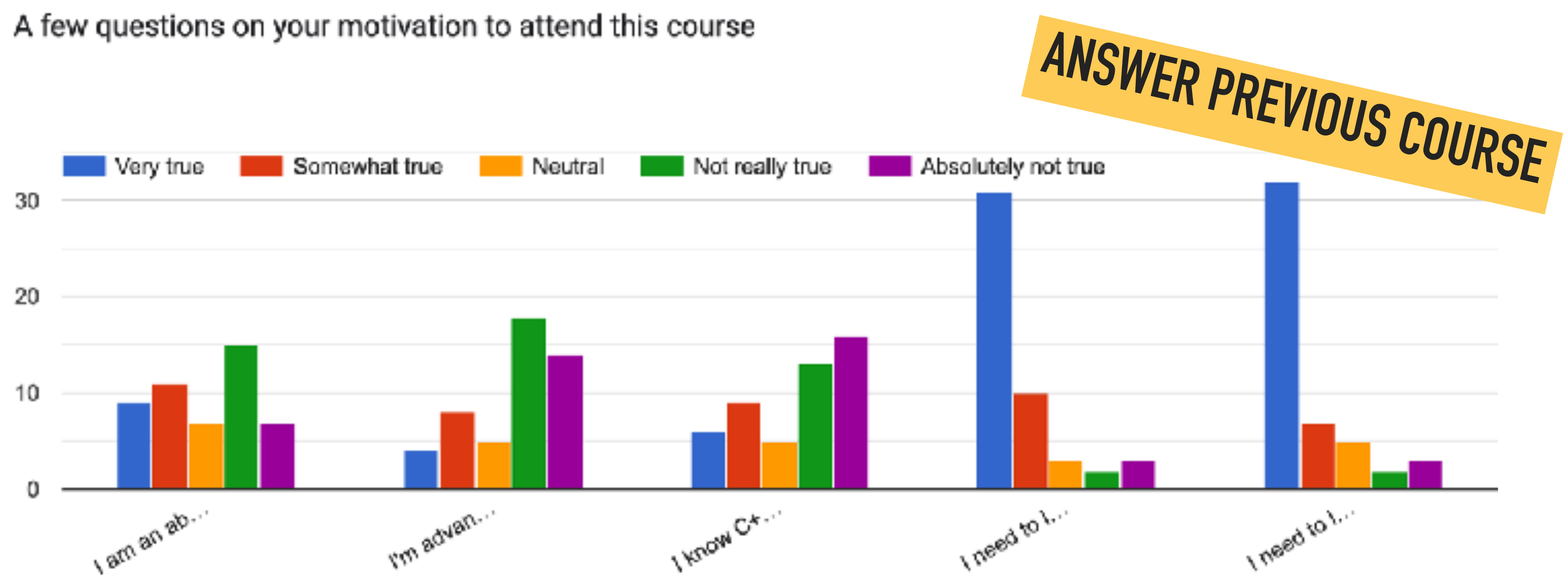
POST TRAINING DISCUSSION

STEFAN ROISER, 16 NOVEMBER 2022

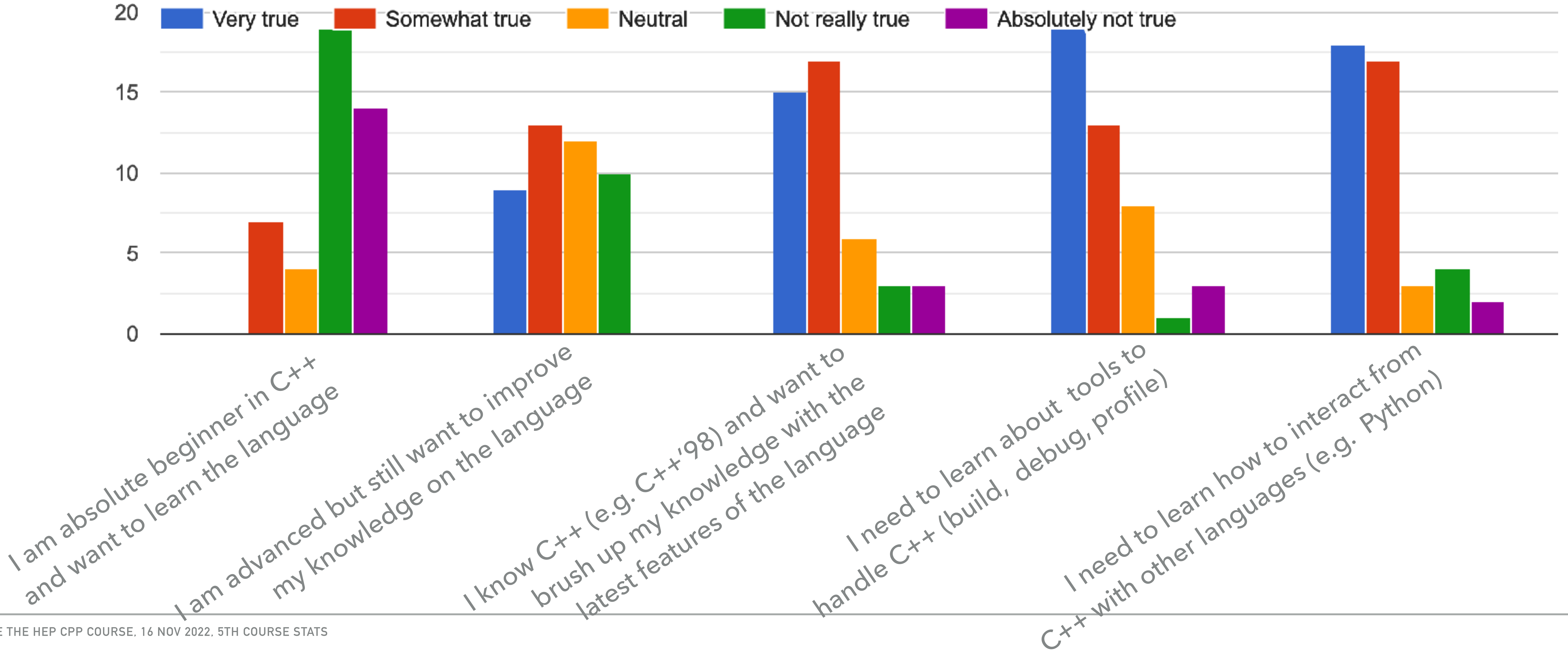
“ADVANCE THE HEP C++ COURSE” MEETING

GENERAL STATS OF THE COURSE

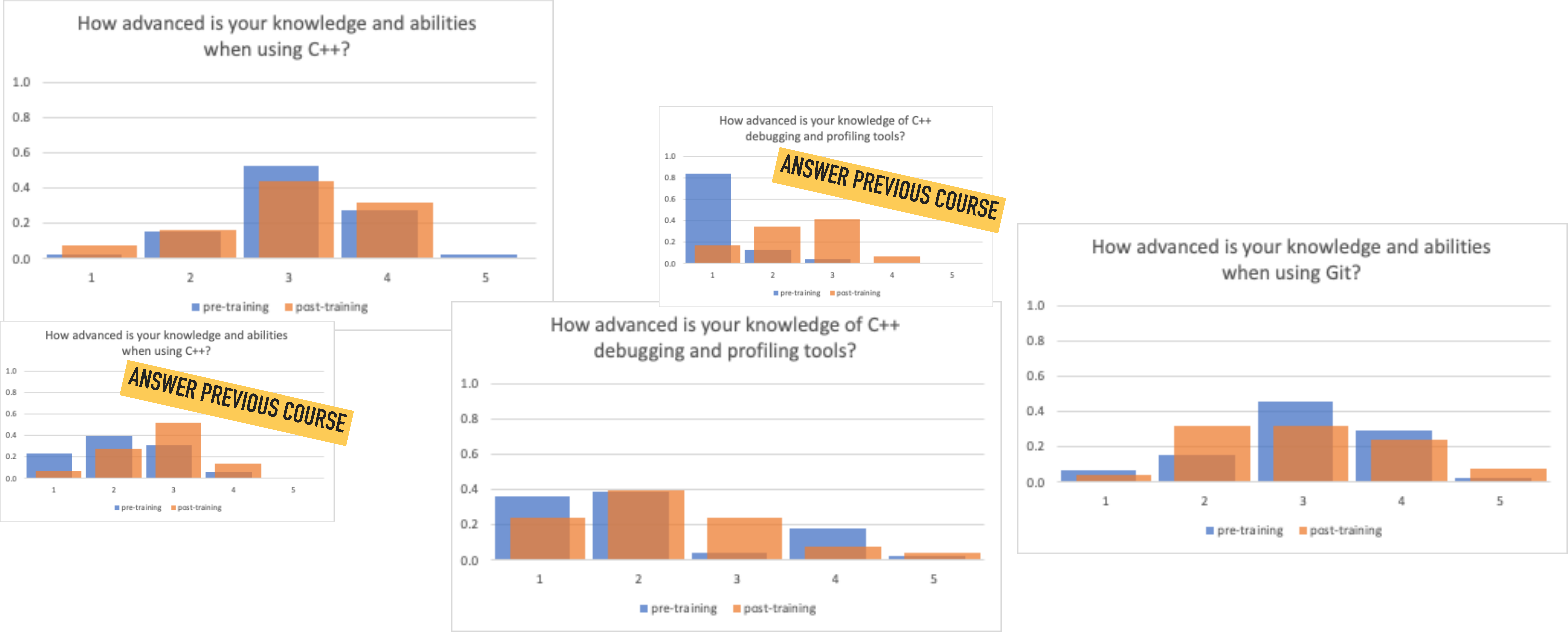
- 100 places opened and booked, within ~ 1 week
    - Around 1 week to fill those first 100, registrations continued for some while
- ~ 40 attendees in lectures and training (IIRC)
    - More absences than in the past –> shall we ask for a little fee (~ 25 CHF)
    - 110 people on the waiting list



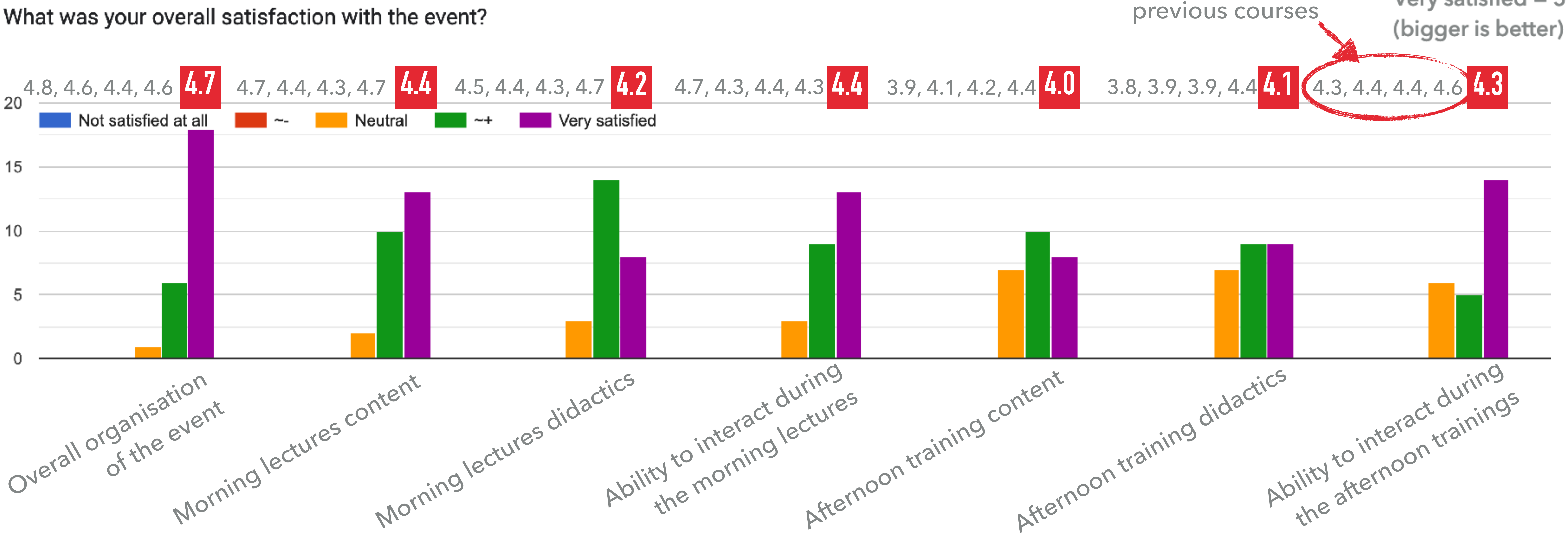
A few questions on your motivation to attend this course



SELF PERCEPTION OF STUDENTS ON COURSE TOPICS PRE/POST TRAINING



OVERALL SATISFACTION WITH THE EVENT



AVERAGE "GRADE"

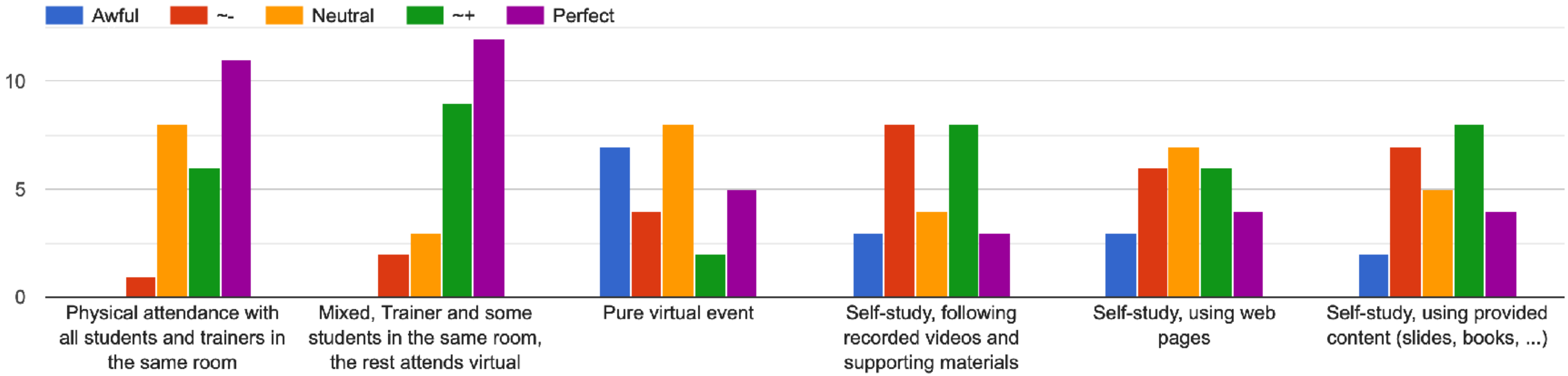
Not satisfied = 1, Very satisfied = 5 (bigger is better)

"grades" from previous courses

"grades" from previous courses	AVERAGE "GRADE"
4.8, 4.6, 4.4, 4.6	4.7
4.7, 4.4, 4.3, 4.7	4.4
4.5, 4.4, 4.3, 4.7	4.2
4.7, 4.3, 4.4, 4.3	4.4
3.9, 4.1, 4.2, 4.4	4.0
3.8, 3.9, 3.9, 4.4	4.1
4.3, 4.4, 4.4, 4.6	4.3

- NB this was the first edition of the advanced course

Please let us know which training format for such events would fit for you best



- Split between “all physical” and “hybrid”

DISCUSSION POINTS?

- Provide a self-evaluation possibility, especially for the advanced course?
- Ask for small registration fee to raise the level of engagement?
- How to handle the attendance easily, especially for online participants? Any tool?
- Check more carefully the HEP affiliation of the registrants?
- Collect all people on the waiting list in a single one? Stay on indico? Sth better suited?

- Content see next slides …

POST TRAINING SURVEY (VERBATIM) TEXT ANSWERS

IN CONTRAST TO PREVIOUS COURSES WE HAVE SPLIT THE CONTENT NOW INTO TWO PARTS ("ESSENTIALS" + "ADVANCED"). THIS WAS THE FIRST TIME WE RAN THE "ADVANCED" PART. DO YOU HAVE ANY COMMENTS ON THIS PART OF THE COURSE?

- Like the essentials course the material is **quite packed!** I enjoyed it, sometimes the only difficulty though is trying to find a part of the slides we've discussed before: the **material does jump around** a bit between sessions. Perhaps a **guide** for this/notes in the indico minutes?
- Sensible to split the course up, otherwise it would be far too long ||| I think it is a good idea. ||| Very helpful and complete! ||| The structure was easy to follow and the explanation was clear. ||| Good to have it separated, considering the amount of content ||| Lot of information in a very brief period of time, but enough to understand bigger sections of the current code. ||| The split was fair - I did not attend the "Essentials" course, but I believe the "Advanced" topics were on the right level.
- Interesting, we may need some more time to go through. 30 minutes more maybe. ||| The slides provide all the information about the new functions in c++17/20. The tutorials on github are comprehensive as well. I think , more time is required , perhaps an a **5 day** training course instead of 3 ||| The level varied a lot ||| I think this has been a great idea. Actually, given the amount of material available, I would even suggest to split the content into **Essentials, Intermediate, and Advanced.** Me and my colleagues found ourselves a bit advanced for the Essentials (especially 1st/2nd day), but we got lost quite a bit on the last day of Advanced ;-) I am sure that an Intermediate course would gather some audience. ||| I did not see the essentials but enjoyed the advanced class. Would have liked to have **one more day** with a slower pace on the same topics (specially when reaching the perfect forward and variadic templates)
- Very interesting but a little bit **too advanced** for me. ||| The first day and a half were very good and possible to follow. The **last part** of the course though was rather **hard** for Essentials-like users.
- Did not take the essentials course so cannot compare them, but for this part it often felt like a lot of concepts were thrown out in the air and then left there without further details (understandable, given the width of the course). Essentially, there was **too much content** to learn it all, as was made clear at the start. Will need time to digest the contents.
- We never managed to fully complete the exercises planned for the day. Would be nice if the **solutions had comments/explanations**, to make it easier for the ones you do alone (by the logic that if you need the solution, there's probably something you didn't manage to figure out, hence an explanation would be nice).

AT THE SAME TIME WE RAN THE COURSE IN A "HYBRID" FORMAT (PHYSICAL ROOM + ZOOM). DO YOU HAVE COMMENTS ON THIS FORMAT?

- zoom side works fine!
- I was in the room so can't comment on the experience of the Zoom attendees ||| I attended in person and everything worked smoothly.
- I think that works. In person is **better commitment** wise ||| For me, personally being in the room **raises concentration** a lot compare to zoom. But good that there's zoom for those not physically at CERN. ||| As a participant who only attended the physical room, I can say that I was very happy with the experience.
- I prefer doing hybrid format since I can **fit my schedule** into the course much easily. ||| I was happy that a zoom attendance was possible, as I **could not attend** in person. ||| Happy with that mainly because i **could not attend otherwise**. ||| I like the hybrid approach, especially for people who are cant attend in person
- In my opinion it is perfectly fine. ||| Was OK for me. |||
- It generally worked well, Zoom participants and mentors suddenly cutting in was sometimes distracting.
- no. I think it's great idea to offer **both physical and remote** presence ||| If possible, please **keep this format** in the future.

IF YOU HAVE A COMMENT ON THE COMMUNICATION TOOLS (INCLUDING ZOOM) PLEASE LET US KNOW.

- The connection seemed to work well; didn't use Mattermost ||| I think the communication worked perfectly fine.
- Thanks for the comment and explanation for the questions.
- The **highlight-style laser pointer was annoying**, it was appreciated when you changed it.

IS THERE ANYTHING ELSE YOU WOULD HAVE LIKED TO LEARN ABOUT?

- Nope! Templating and working with threads were both covered :)
- Maybe more time on **performance measuring and debugging**. ||| GDB ||| A bit on **debugging tools**, but the material available is great for self-learning. ||| Would be great to have more exercises on the debugging tools and profiling
- I would also have preferred more of a **focus on C++20**... e.g. if there is a way of doing something in C++20 that improves on previous versions of the language, discuss C++20 preferentially to avoid running out of time ||| I am **looking forward to the seminar on C++ 20.**
- Although very limited in time, my suggestion would be to add a small **section on Testing** - especially googletest - http://google.github.io/googletest/advanced.html. ||| For example what implications (pros/cons) would come from using either in testing and maintainability in a **ci pipeline**.
- Actual **experiment-specific C++ frameworks** as an example (how the concept is applied as an example) ||| I'd like to learn more about things related to scientific computing but not related to the core C++ language, such as **Cuda, MPI and Boost.**
- **tuple**
- Large programs and the interactions between different limbs of a software suite (ie more about large scale structures), eg how dependencies within different parts of a program relate to each other ("I know that my collaboration uses C++17, but there are features within C++20 I want to use -- which parts can I write in my preferred version without having to commission a transfer for the entire project")
- **cmake**
- maybe a bit more on the **performance evaluation** of a program and optimal programming (like good memory handling, vectorization, ...) . I know the time is limited so if it is not possible to extend the time of the course, i think it's ok like this in terms of content coverage.
- I would have liked to see **more on templates** in general than variadic templates but this is most likely just an a lack of my knowledge in general. Also see more applications of when would it be better to use templates over object interfaces overall. Would have also liked to see more in depth RAII concepts and more example applications.

DO YOU HAVE OTHER COMMENTS CONCERNING THE CONTENT OF THE EVENT?

- My main comment concerns the **exercises**. I think it would be better to have a **pure focus on the topic** in hand for each exercise and make everything else as simple as possible. An example is the move semantics exercise which used an over-complicated setup, whereas a simple class with a single integer data member could have demonstrated the same thing with much more clarity. Similarly the tuple example was probably more complicated than necessary to demonstrate basic variadic templates; it could possibly be a "part 2" exercise with "part 1" doing simpler exercises. ||| Some questions could be clarified, improved. E.g. **variable names are sometimes confusing**
- I loved it. I will refer to the slides in my work!
- Very good. But, it would be nicer if you can actually give examples related to HEP experiments-related C++ codes (I saw a few, which was very helpful).
- We could have done **3 hours exercises** sessions in the afternoon instead of 2.
- They're definitely oriented towards advanced users. I think there is a gap in experience between people with knowledge such as described in the essentials and people able to follow the course with profit.
- Would be cool to have **both courses twice per year,** this way won't be necessary wait a whole year to attend to other course.
- I enjoyed the **debugging/tools topics (sanitizer, etc..),** I feel these are very useful tools to see. Also enjoyed the **concurrency topics** (mutex, amotic, etc...). Enjoyed also the standard library usage examples.

PLEASE LEAVE US A COMMENT ON ANY OF THE POINTS ABOVE IF YOU LIKE. (GENERAL COURSE FEEDBACK)

- The course is **super intense but I don't mind** it. It's nice to have an **instructor** for the **workshop** time, but having the solutions to refer to is also great for my understanding. Rather than using understanding to gain the solution sometimes I use the solution to gain understanding and know what question to ask the workshop volunteer :)
- In some cases the **material was rushed**; probably future iterations will improve on this as the lecturers get a better idea of how much time to spend on each topic. Also see my comment on the exercises… ||| **Afternoon contents needs way more time than scheduled**. **Lecturers did a great** overtime job, but I believe this should be foreseen in schedule
- I would suggest having **unit tests for more exercises**, especially because many of them are left to be done at home.
- For me, the **exercises for one single topic where too long**, I would have prefer to have 10 minutes to solve the first part, then to have the correction then to go to the next step. To make a long story short I would have prefer to have some more guiding during the training. But it was great really !.
- A simple **list with the exercises for the different days** would be nice.
- The level of people differed quite a lot and people were shy to ask simpler questions. Maybe some **"check your level quiz"** could be provided.

IF YOU FELT UNPREPARED AND CONFUSED AT ANY POINT, PLEASE ELABORATE ON WHAT WE COULD HAVE DONE BETTER.

- I had to do some very weird debugging of permissions and certificates to get gdb running on Mac OS, but that is not related to the course. I think it might be handy for people not running on linux to have a big checklist of tools to check they work before the week starts. I think there was also a point where the instruction was to look for some output from the debugger and my MacOS debugger said it doesn't do that one, and i just had to move on.
- Please see my comments on the exercises; I think they need to be more focused on the relevant topic with less extraneous material
- We needed **valgrind**, hence please **default us to using lxplus** rather than personal laptops.
- I have not used the language for a long time except just reading some scripts whenever I need to. I have seen many features for the first time.
- I **studied** the material of the **Essentials** course on my own. I was **able to follow** the Advanced course pretty well, except for the Expert C++ part.
- I attended the beginners course in 2017 or so, I don't use C++ in my work, but sometimes have to read ROOT macros or DaVinci/Brunel C++ scripts. For this course, I read through half of the slides the day before, which helped to remember the basics, but day 3 was too deep for me (purely my problem, I think the content was great for people who followed)
- I just wanted to say that for the advanced concepts about which you hear for the first time, this is **not really enough to truly understand** them. I took it more as learning that this exists and roughly what does it do, and if I really want to learn about it I'll look for more dedicated material (eg lecture videos). If you provided such optional longer lecture videos on a given topic, that would be of course amazing, but perhaps it's asking too much of the volunteers.
- The course goes into **very advanced** (and new!) features of C++ very fast. I was quite comfortable on the first day, but on the third I found myself totally lost. Given the amount of material available, I'd consider seriously dividing the content into three levels. This may result into shorter "morning" lectures, but this may allow more time for hands-on training.
- I felt a bit confused during **"More STL"** lesson . Mainly because i never used those tools, i could not understand the usefulness of that. The exercise helped a little in that sense
- Not because of the event but for **timings**: wait until march to attend the basic topics is too much time and advance topic was too much for start with it.
- The topics of **perfect forwarding and variadic templates was too advanced** for me to follow through. I understood the overall use (thanks for the final examples) but much of the details I could not follow in such a short time. Personally, the **move semantics was a bit confusing** (where should I use it exactly) but still understood after some time. The **valgrind chart is confusing to understand**, in particular what do the syntax of the functions mean and how to properly identify which do they relate to in my code.

PLEASE PROVIDE ANY OTHER FEEDBACK ABOUT THE ORGANIZATION OR EXECUTION OF THE WORKSHOP FOR US TO CONSIDER IN THE FUTURE.

- I really enjoy these workshops. Thank you.
- Organisation was great; as per my previous comments the content needs a bit of refining w.r.t. focus on new features and streamlining the exercises. You might also consider whether an **extra day** would have been worth it. Most likely **many of the attendees would have been at CERN for the whole week** anyway. However, I appreciate that it is a great deal of work to put together something like this, and one would pay thousands in the commercial world for a similar course, so thank you!
- I would e**xtend the course to more days - maybe for 4 or 5**. Pace was great, and I loved being able to ask many question, but more time would allow to discuss the even harder topics which we needed to skip this time.
- The actual examples from ATLAS or CMS C++ codes will be more beneficial to understand how the concepts are applied in the HEP experiments.
- I really liked the course. Thank you all for the time dedicated to it.
- Many thanks to all of you ! it was really (really) interesting !
- In my opinion everything was fine.
- Very much enjoyed the course and the content, Excellent job by the team ,and looking to join more courses.
- thank you, this course is on a great trajectory!
- Really appreciate the online resources! Would have liked to attend, but the slides are still great.
- Since it's an HEP programming course, i would like to see **more particle physics example** in the applications. But how it's done at the moment it's not limiting of course