CppIndiaCon 2023 - Call For Speakers

Unlock the stage, unleash your C++ knowledge!

The submission will take approximately 9 minutes to complete.

Presenter's Information

1. Name *

   As you would like it to appear in the conference program.

   Ankur Satle

2. E-mail *

   For conference use only. We will not share this with third parties except as required to run the conference.
   We strongly recommend using a *personal* email address because employment situations change, and companies' spam filters sometimes reject our messages to you.

   ankursatle@gmail.com

3. Contact

   We are only reaching out for a quick response and assure you no spam.

   +919970187488

4. About You *

Please use the third person, avoiding "I" or "me."
Use your full name (once) in the first paragraph and your first name once in each subsequent paragraph. Otherwise use your pronouns.
Each submitter will be given the opportunity to review and submit edit their conference bio to include a photo, website, and public social media contacts.

https://www.cppindia.co.in/conference/2022/speakers/ankur/

5. Photo (head shot) *

Please enter the URL of a headshot appropriate for the conference program.
If you leave this blank, we'll use what we find on the net and/or nag you ceaselessly to supply a photo.

https://www.cppindia.co.in/conference/2022/speakers/ankur I would like to update this later :P

6. LinkedIn Profile

https://www.linkedin.com/in/ankursatle/

7. Twitter Profile

https://twitter.com/AnkurSatle

8. GitHub profile

https://github.com/sankurm

9. Personal website

https://ankursatle.wordpress.com/

Submission Info

The information in this section is directed to attendees and must be suitable for the conference web site.

10. Talk/Presentation title *

Value-Oriented Design - by example

11. Talk abstract! *

As it should appear in the program. About one to three paragraphs.
This is your pitch to both the Committee and to potential attendees about why they should see your presentation.
Try to answer the reader's questions, ***what will I learn?*** and ***why is this important to me?***

Value-oriented is the best, concise, managed way to implement in C++ for most cases. We will discuss in this session how we can unlearn old ways of manually doing things & costs that object-oriented approaches levy on us with simplicity, correctness, compiler-generated ways of value-oriented programming. The practical examples are intended to drive home this approach in

12. Talk outline (topics coverage, flow and content. This feeds to the selection process) *

Coverage

1. Intro - example of a piece of code dealing with values contrasted with one without
    a. Highlight manual management
    b. Discuss lifetime
    c. Contrast ownership
    d. Emphasise optimization
2. Dealing with persistent objects - those that live beyond their instantiation scope
3. Data-structures with value types
4. Class abstractions built with value types vs those built without
    a. Overheads of explicit implementation - correctness, maintainability, obsoletion, how more sticks to this like velcro
    b. Same example with value types
    c. Rule of 3/5/0 in action
5. Instantiation via factory class/methos
    a. Guaranteed Copy elision
    b. Move
6. Passing value types - Cover optimizations of
    a. Temporary types
    b. std::forward
    c. Help from move
    d. Objects that aren't even instantiated in main memory
    e. Const - & related optimizations
    f. (possible discussion, if it fits) Passing large objects vs. parts - at the right-level of abstraction
7. CRTP & possible performance gains
    a. CRTP vs virtual
    b. LinkedIn had a post with an image of performance benchmarking many approaches including CRTP - it was winning
8. Passing Callables around
    a. Plain passing
    b. std::function & related types
9. Working with many types
    a. std::variant & overloaded visits
10. Type erasure
    a. std::any
    b. Boost?
11. Working with errors & failure
    a. C++17 std::optional
    b. C++23 std::expected
12. Working with values in a concurrent environment
    a. Sharing & Ownership
    b. Fire-and-forget
    c. Chaining
13. Value Template Parameters - (exploring if this fits)
14. Value-oriented programming with Carbon or cpp2 or Rust or Val, if possible

13. Session Length *

Standard sessions are 60 minutes. Preferred, min, and max can all be the same value.

- (●) 60 minutes
- ( ) 45 minutes
- ( ) 15 minutes (Lightning Talk)

14. Attendee expertise/knowledge level expected

- (●) Novice
- ( ) Intermediate
- ( ) Expert

To the organizers

The information in this section is not published on the website as part of the conference program but is intended only for the Program Committee and/or other conference organizers.

15. Video URL

Please include the URL of a video of you giving a technical presentation (as similar as possible to your proposal).
We do accept submissions from first time presenters, so this is optional, but if you have a recording of yourself presenting, it helps the PC evaluate your presentation abilities.
You may use a video that you've made and posted yourself. If no such video exists, consider a link to a slide deck that you've created.
Feel free to add any comments you feel necessary about this material.

https://www.youtube.com/results?search_query=ankur+satle

16. History

If you've presented *this submission* before, please tell us where and when.
We *do* accept submissions of sessions presented at other conferences.

It's a CppIndiaCon First!

17. Session Material

List material that you will submit to the public conference repository: (slides, source code, etc.)

Slides, possibly Video, Code

18. Comments

Anything you'd like the Program Committee to know about your talk.
This would be a good place let reviewers know your pronouns.
If you are making multiple submissions, but don't want all of them to be accepted, please let us know your limit and if you have any preferences about which are accepted.

Jai Hind!

19. I grant permission for my session to be recorded for public posting. *

I Agree

20. CppIndia can utilize the information you provided for the community's benefit. *

I Agree

21. The content you submit is your own intellectual property or you have obtained permission to share in the CppIndia community. CppIndia will not be held accountable for any conflict of interest.
*

I agree