



CElite Coding Challenges

Instructions, Rules, & Guidelines

Created by SadMafioso

Table of Contents

1. Project Objectives.....	2
2. Rules and Guidelines.....	5
3. Submitting Your Solution	
4. Github Tutorial	
5. Scoring	
6. A	
7. A	
8. A	
9. A	
10. A	
11. Aa	
12. A	
13. A	
14. A	
15. A	
16. Aa	
17. A	
18. A	
19. A	
20. A	
21. A	
22. A	
23. A	
24. A	

Project Objectives

I believe that the best way to learn real world programming is to work on projects, and take them as far as you can, figuring out the little quirks of your language and little optimizations along the way. While in school I've personally mostly worked on static and controlled projects where there is basically an expected solution already, I don't think this is the optimal way to learn programming. It's your process, from taking a problem from words and turning it into a full fledged program, that is the most important factor. Throughout this process you will most likely need to flex your Google-fu a bit and find relevant information through online resources. You might be working with other developers or on your own, but the fact always remains that reaching out to others for help is almost always in your best interest. No programmer is able to bust out a working application for a specific problem without some outside reference - StackOverflow, documentation, random blog posts, Youtube tutorials - and it's your skill in translating this information into a working solution that makes you a true programmer.

The CElite Coding Challenges aim to help develop these skills in our community, and share our collectively gained knowledge to the world in hopes that it may help at least someone else in the future. We will pose problems to the community and invite them to create and share solutions. It has not been decided completely, but each challenge will take place for about a week. Each challenge will be divided into three difficulties to allow people of all skill levels to submit a solution. At the end of the time period, all submissions will be received, and the CC Team (Coding Challenges Team), with the help of community judges, will look over each solution and make notes on its effectiveness, efficiency, readability, and other coding standards. Several solutions

will be chosen to be reviewed on a livestream, based on several factors. Does the solution contain a common mistake seen throughout several other submissions? Is the solution exceptionally well written and a “role model” for other solutions? Which solution will help the most people if reviewed on stream? There are many considerations to be made when choosing the right solutions to review, which will be up to the CC Team to figure out.

A website will be set up to receive submissions. You may work by yourself or with a group, and groups are highly recommended! When you work in the industry chances are you will have to work with a team, peers with different skill sets and specializations, all putting the work in to create a solution polished in every aspect imaginable. Meet some new people, have a voice call with them, discuss your goals, your individual strengths, and what you can each contribute to the project. If you decide to go solo that is completely fine - just know that you may be missing out on the full experience of software development we want to facilitate with this project.

The CC Team and judges will also pick the top (three?) solutions for each difficulty as the winners. The number one solution will be saved on Github as the “finished product” for the challenge’s repository, most likely with small changes made based on other solutions or community feedback. Every other solution will be added to the documentation for the repo, and so each participant will have the added incentive of showing off their contribution on their graph. Theoretically we will be crowdsourcing small applications with useful purposes, and you will be contributing to open source software.

Once all submissions are received and solutions are chosen for review, the stream will take place. The CC Team will contact other community members and judges with specializations in a certain language or a type of project to create essentially a “lesson plan” for the

stream with specific examples from submissions. We will also be using the website <https://repl.it/> to allow for a live demonstration of the code. Through the stream we will examine common mistakes, areas to improve, efficiency, etc. in order to gain a deeper understanding of the problem and why we chose the winning solution.

Each solution will be archived into the Github profile - <https://github.com/CEliteCodingChallenges>. When you make a submission we will ask you to include a raw text file of all of your work (format in Chapter ?) that will be included in a documentation file for the winning solution. Github does not work in a way where we can see everyone's solutions at the same time, and so pushing in a text file of your solution will act as your contribution to the repo - and it will be contribution! Your solution could have aspects of it that are included in the featured solution, which will theoretically be an amalgamation of all of the best aspects of the submissions.

This is by the community for the community. I really want to create a good resource to help out my fellow programmers, and so I want to be completely transparent about the process behind this. I am always open to critique, suggestions, comments, and questions. I have no desire in this other than to help develop myself as well as my peers on the Discord server, and anyone who might stumble upon our content. I want to record the streams and put them up on Youtube, with a description on what is being covered and timestamps pointing to different topics within the video. I want to transcribe these streams into documentation that could be of use to people working on their own projects. I want to help as many people as I can while directly helping others in the Discord.

Rules and Guidelines

1. All participants agree to be respectful of others – insulting, slandering, or otherwise putting down any other participant because of their solution or for any other reason is not allowed and action will be taken accordingly.
2. All submissions must be received within the given deadline. Late submissions will not be taken into consideration when choosing the winning solutions and what will be reviewed on stream.
3. Participants may work by themselves or as a group. No formal registration of groups is required, submissions must simply include the names of all group members.
4. Always provide some form of documentation throughout your solution. This will help the CC Team understand your solution and follow along when presenting it on stream. Groups should also make notes of where each member made contributions.
5. Not all solutions will be reviewed on stream, depending on submission volume and time. While we would love to help each and every person in the entire world with all of their problems, we simply don't have the time. Priority will be given to reviewing solutions that contain common mistakes and design flaws and comparing them to more effective solutions.
6. Participants may provide solutions for one, two, or all three of the problems, but in most cases only up to one solution per participant will be reviewed on stream.

Notes

- You are free to join the stream to explain your solution, though you are under no obligation to do so.
- While repl.it is a very useful resource, it will not be able to compile full fledged programs with GUIs. For this reason, we will begin with simpler programs that are complex in their execution but are interacted with a console. As we progress to more complicated programs, we will find a way to be able to examine the code and still run it for the stream. It may be preferable in some instances to include screencaps or videos of your program working that we can reference while looking at the code.
- Judges will be selected based on knowledge and availability. While the community as a whole is encouraged to share their thoughts, the judges will get the final say on scoring the solutions.
- The end all philosophy of this is to learn together. There may be solutions that are too easy for you, or those that are too difficult. If that is the case, add a feature, or team up with someone that is a bit more knowledgeable with that topic. Use this as a resource for your own learning and understanding of programming. If your solution wasn't great, don't get frustrated! Join the stream and we can analyze why a different solution worked better, and you will have that knowledge for next time.