COPS

Club Of ProgrammerS

Talk on Google Summer of Code

Topics to cover

1. Open Source
2. Contributing to Open Source
3. Open Source programs
4. Google Summer of Code

Open Source

Open source projects, products, or initiatives embrace and celebrate principles of:

- collaborative participation
- rapid prototyping
- transparency
- meritocracy
- community-oriented development

Open Source Software (OSS) is software with source code that anyone can inspect, modify, and enhance. OSS is **peer-reviewed**, transparent, and **reusable**.

Open Source doesn't mean that it's economically free to use.

Why is it important?

"In real open source, you have the right to control your own destiny."

- Linus Torvalds

Imagine a world without open source...

- There would be no C, PHP, Python programming languages, no Linux (which has the maximum share of servers in cloud) and no GNU project.
- We'd live in a world of Symbian or Windows phones and market of PC operating systems would be almost wholly owned by Windows.
- We would miss Android, Apple's iOS and macOS because they wouldn't have the open-source kernel to build their OS on.
- There would've been no innovation, no (or maybe slow) progress in the technology landscape.

Contributing to Open Source

Open-source projects uses one of the various Version Control Systems, like GitHub, GitLab and BitBucket.

Contribution helps build one's GitHub profile, which helps to showcase your software development skills.

Various facets of contribution:

- Contribute code
- Contribute to documentation
- Do code reviews
- Find new issues

Open source programs

https://github.com/tapaswenipathak/Open-Source-Programs

These programs are meant to promote the culture of open-source amongst technical students. Students contribute to real OSS projects under the guidance of a project mentor.

Famous programs are:

- GSoC
- Outreachy

Takeaways from participating in them:

- Learn real-world software development skills
- Work on various technologies
- Be a part of a transparent community driving the project
- Various rewards like stipends, swags, or prizes.

Google Summer of Code

https://summerofcode.withgoogle.com/

World's biggest open-source program, in which students work with an open-source organization on a 3-month programming project during their summer break. The selected students also receive a stipend!

How to apply?

Students contact the mentor organizations they want to work with and write up a project proposal for the summer.
If accepted, students spend a month integrating with their organizations prior to the start of coding.
Students then have three months to code, meeting the deadlines agreed upon with their mentors.

Three simple steps to crack GSoC

Simple != Easy

1. Get involved in the organization (start from Nov/Dec)
   a. Use their product (by building it from source yourself)
   b. Reading their Contributing guide, their documentations
   c. Join their mailing list/IRC/Gitter and interact
   d. Start by solving minor issues
2. Draft your proposal (Feb - April)
   a. Maximum of 3 project proposals can be submitted.
   b. It's important to understand the project and its requirements
   c. Communicate with project's mentor beforehand
3. Stay focused during the coding period (May - Aug)
   a. Manage time
   b. Engage in regular communication with your mentors
   c. Meet your proposed deadlines

Useful sites

Open Source contributions

- Awesome lists
  - https://github.com/MunGell/awesome-for-beginners
- https://up-for-grabs.net/
- https://www.codetriage.com/
- https://github.com/trending

Google Summer of Code

- https://google.github.io/gsocguides/student/
- https://medium.com/coding-blocks/google-summer-of-code-211ae61a8edb

Important Tips!

- Find an experienced software developer in the community, listen and engage with them to incorporate their software development learnings.
- Open-source community is there to support you, so don't hesitate in sharing an ugly code or asking stupid questions.
- Practice git, GitHub etc. It's important to understand the git workflow to do a meaningful contribution.
- Practice best software development practices, like using human-readable variable names, commenting generously, writing tests, using CI etc
- **Be receptive to constructive criticism**

- **Don't be afraid of the sudden learning curve you face.**