



Apprentice Software Crafter

Work in an exciting environment on software projects from NGOs to Banks, from IoT to Cloud, with a culture that focuses on quality, teamwork, and individual growth.



Do not print this document.
Don't waste paper.
Save the environment.

Who we are

Nelkinda Software Craft, founded in 2015, is a culture leader in quality software development. Through our project and coaching work, we are advancing the way by which organizations develop software. Our clients range from startups to multinational banks. Having organized, hosted, and supported more than 100 events in the software development community over the past 5 years, we are one of the most active contributors to the international software craft scene.

Job Brief

Industry	Computer Software, Training, and Consulting
Occupational Category	Software Developer (Junior)
Employment Type	Full-time employment, 40 hours per week
Job Location	India/Pune/remote
Date Posted	2021-02-24
Probation Period	6 months
Join Date	Immediate/ASAP

Job Brief Description



Angel says: We also accept applications from applicants who do not satisfy all the aptitude requirements but have the right attitude to quickly upskill themselves.

Who you are

- You have 0-2 years of work experience and know at least one of the following languages: C, Clojure, Go, Java, JavaScript, Kotlin, Lisp, Scala, or TypeScript extremely well.
- You have a passion for solving problems and providing not only workable but maintainable solutions.
- You understand that learning is a never-ending journey. Because of that, you keep in touch with the latest improvements of software engineering by reading books and blogs, following industry leaders on social media, and watching videos. You support team members on their learning journeys.
- You have strong analytical and reasoning skills based on the scientific method with an ability to structure and visualize processes and outcomes.
- You have outstanding communication skills in English and the ability to work collaboratively.
- You have a degree in software engineering, computer science, mathematics, electronics engineering, or a related field, or equivalent work experience.
- You have started to learn structured, object-oriented, and functional programming, and want to learn design patterns, the SOLID and package principles, and the 4 rules of simple design.
- You want to embrace change and harness it for competitive advantage.
- You are eager to learn new tools and technologies.
- You do not judge tools and technologies by their age but by their usefulness and fitness for purpose. You can use old and new tools and technologies alongside each other to combine their strengths.

What you do

- You design, test-drive, and write clean code, working on a variety of languages, technologies, and architectures.



Responsibilities

- You contribute to our projects as a junior engineer, where you learn through participation and observation about the design and development of software using Extreme Programming and related practices like simple design, pair and ensemble programming, test-driven development, behavior driven development, continuous refactoring, continuous integration, trunk based development, and release to production several times a day.

What we offer

- The opportunity to work with a large variety of projects, clients, and technologies.
- A health insurance plan with family coverage.
- A detailed individual growth plan and support to help you become a master software crafter.
- An allowance for books, internet, conferences, and training.
- An environment that allows you to grow and shine as an individual in the international software craft community through our various event platforms.
- A value system that ranks principles and practices higher than tools and technologies.
- Opportunities to contribute to open source projects.

How to apply

- Send your application letter and résumé in PDF or XHTML5 format to career@nelkinda.com.

How we interview

- 60 minutes video call on Google Meet or Zoom.
- Offline coding assignment plus 60 minutes video call in which you have to explain your solution to the coding assignment.
- 120 minutes remote pairing session on Discord, Google Meet, or Zoom.



How to prepare for the interview

Make sure you have your laptop ready with a good webcam and microphone or headset and that video calls work, the audio is good, and you know how to share your screen. Test with a friend.

An interview is a mutual activity, prepare a list of questions you would like to ask.

- Behavior Driven Development <https://www.agilealliance.org/glossary/bdd/>
- Cucumber <https://cucumber.io/>
- Extreme Programming <https://www.agilealliance.org/glossary/xp/>
- Flaccid Scrum <https://martinfowler.com/bliki/FlaccidScrum.html>
- Git <https://git-scm.com/>
- GitHub <https://github.com/>
- JUnit 5 <https://junit.org/junit5/>
- Manifesto for Agile Software Development <https://agilemanifesto.org/>
- Manifesto for Software Craftsmanship <https://manifesto.softwarecraftsmanship.org/>
- Principles of Object-Oriented Class Design <http://butunclebob.com/ArticleS.UncleBob.PrinciplesOfOod>
- Refactoring <https://refactoring.com/>
- Scrum <https://www.agilealliance.org/glossary/scrum/>
- Specification by example https://en.wikipedia.org/wiki/Specification_by_example
- Test-Driven Development <https://www.agilealliance.org/glossary/tdd/>
- The Four Rules of Simple Design <https://martinfowler.com/bliki/BeckDesignRules.html>
- The Three Laws of TDD <http://butunclebob.com/ArticleS.UncleBob.TheThreeRulesOfTdd>
- The Transformation Priority Premise <https://blog.cleancoder.com/uncle-bob/2013/05/27/TheTransformationPriorityPremise.html>
- Trunk Based Development <https://trunkbaseddevelopment.com/>