CRED - interview process

Introduction

We will be conducting the interview process virtually. The candidates are requested to ensure that they have optimal internet connectivity for a seamless experience during the interview rounds.

We will be conducting the following interview rounds as described below

Interview rounds

R1 - coding round

The coding round will be conducted on the Hackerrank platform. Please familiarize yourself with the Hackerrank platform to ensure that you are comfortable with the online setup along with the programming language of your choice (along with the version Ex: Python3, Java 8, etc).

This round would comprise 3 questions and a duration of 90 minutes. The questions would range across topics of **problem-solving**, **leveraging data structures & algorithms** to solve the problem elegantly.

Skills: Problem-solving, Data Structure, Algorithms

FAQs on Hackerrank as a platform

1. How to debug code
2. How to write code
3. How to submit my code
4. More FAQs

PS: Don't use mobile or earphones during this round.

R2 - project round

The project round would be conducted on the Hackerrank platform (project module) and would have a duration of 150 minutes (2.5 hours). The intent of this round is to gauge how well you can contribute to a real-world problem and a structured codebase.

The expectation during this round from you is to understand the problem statement (real-world and ambiguous), able to break down and implement the same through the code.

We will also be providing basic code structure & skeleton to avoid cold start problems. You will be expected to navigate through the code and understand the current setup before you could start adding the implementation at the appropriate places.

Skills: Hands-on Coding, Solving real-world problems, Clean Code, familiarity with OOPS methodology (Encapsulation, Abstraction, Modularity, Extensibility), SOLID, DRY principle.

You will be evaluated on the basis of the above points and functional completeness of the project.

The choice of programming language would be limited in the project round to the following

- Java 8
- Python 3.5+
- NodeJS
- C++

Prerequisites for the project round

Across the programming languages, we suggest setting up IDE locally for ease of use, convenience, and productivity.

Java

- Java 8 (setup locally)
- Maven (mvn) for build & running test
- Junit5 framework for unit test
- IDE
  - Online IDE (VSCode online) is supported on the Hackerrank platform
  - Local IDE - IntelliJ IDEA, Eclipse, or Netbeans

Python

- Python 3.5+ (setup locally)
- Install requirements `- pip install --user -r requirements.txt`
- IDE
  - Online IDE (VSCode online) is supported on the Hackerrank platform
  - Local IDE - PyCharm, VS Code, etc.

NodeJS

- nodejs
- npm
- npm test to run the test cases
- IDE
  - Online IDE (VSCode online) is supported on the Hackerrank platform
  - Local IDE - WebStorm, VS Code, etc.

C++

- C++ Compiler - GCC-9/Clang-11

- cmake v3.20
- Boost 1.75.0
- IDE
    - Local IDE - VS Code (with CPP extension), CLion

PS: Don't use mobile or earphones during this round.

R3 - technical round

The technical round would be conducted virtually over Google Hangout or Zoom for a duration of 75 minutes.

During this discussion, you will interact with the CRED team members and the topics would range from problem-solving, applying first principles to break down & solve them, leveraging data structures, algorithms, and CS fundamentals along with domain modeling of real-world problems.

PS: We promise to not ask standard questions like "Find a loop in Linked List" and make this an interesting discussion

R4 - CRED Mark

The CRED Mark discussion would be conducted virtually over Google Hangout or Zoom for a duration of 60 minutes. In this discussion, you will interact with senior tech team members and leaders.

This conversation would be centered around you and we would like to know and learn more about you - your behaviors and values & belief system that has shaped you. Along the way, we will also discuss how that resonates with the CRED values.

PS: No questions about "where you see yourself in 5 years" from us

Resources & Links

- Sample code - https://github.com/gauravrmsc/calculator
- 42 IntelliJ IDEA Tips and Tricks
- How to write clean code? Lessons learnt from "The Clean Code" — Robert C. Martin
- Clean Code - Uncle Bob - Playlist
- google/libphonenumber - C++
- Git basics
- Git Basic Commands