Pathways Program

Mentor Guide

Updated 5/17/23

Thank you very much for serving as a mentor! The mentoring component of our program is ***crucial*** to developing effective entry-level software developers. This document provides a guide to what you *might* do with your mentee throughout the program .

Stage 1 (5 weeks)

- Overarching goal: Develop a solid *mentor/mentee* relationship
- Time requirement: One hour a week via phone or virtually or a short meeting (such as a lunch)
- Possible topics of discussion:
    - Advice from mentor on characteristics of an effective developer
    - Introduction
        - Team information - members, code base, tech stack
        - IDE/languages used at the sponsor (possibly with short demo)
        - Source/version control used at the sponsor (possibly with short demo)
        - Development process at the sponsor
        - Best practices/conventions used at the sponsor
    - Summary of Pathways Program main topics that might need review with mentor
        - Week 1: VS Code, control structures, methods
        - Week 2: Arrays, text files, source control, JSON, algorithms (search/sort)
        - Week 3: OOP (encapsulation, inheritance, polymorphism)
        - Week 4: Abstract methods, interfaces, other data structures (lists, dictionaries)
        - Week 5: Challenge (participants should receive minimal guidance with this)
- Regular activity and milestones:
    - Weekly conversation
    - Create an individual professional development plan to fill the high-priority gaps between the sponsor tech stack and what is used in the Pathways Program

Stage 2 (5 weeks)

- Overarching goal: Develop a solid *team/participant* relationship
- Time requirement: Practicum - Friday afternoon at sponsor organization
- Possible topics of discussion:
    - More detail
        - Team information - members, code base, tech stack
    - Frameworks used at sponsor organization
    - Software patterns used at sponsor organization
- Summary of Pathways Program main topics that might need review with mentor
    - Week 1: Front-end - HTML, CSS, Javascript, Typescript
    - Week 2: Front-end - Javascript, consume APIs (GET), Bootstrap, Angular
    - Week 3: Back-end - Software goals, principles, patterns, layered architecture (MVS), Inversion of Control/Dependency Injection, SOLID, unit testing
    - Week 4: Back-end - APIs w/ MS Web API and Entity Framework, APIs with NodeJS/Express
    - Week 5: Challenge (participants should receive minimal guidance with this)
- Regular activity and milestones:
    - Shadow different members of the team

    - Attend stand-ups and other team meetings as available
    - Review, update, and work on the individual professional development plan

Stage 3 (5 weeks)

- Overarching goal: Learn the team's workflow and tools
- Time requirement: Apprenticeship - Friday at sponsor organization
- Possible topics of discussion:
    - More detail
        - IDE/languages used at the sponsor (ideally with a hands-on experience)
        - Source/version control used at the sponsor (ideally with a hands-on experience)
        - Best practices/conventions used at the sponsor
- Regular activity and milestones:
    - Attend stand-ups and other team meetings as available
    - Work on the individual professional development plan
    - Possibly complete a simple, level-appropriate story for a small feature or fix
        - Create and task out work
        - Run company software. Debug. Run solution.
        - Write automated tests in the solution
        - Manage source code in the solution
        - Create a Pull Request for a small feature or fix
        - Complete a code review in the solution for the PR

Stage 4 (5 weeks)

- Overarching goal: Complete at least two level-appropriate stories
- Time requirement: Apprenticeship - Thursday and Friday at sponsor organization
- Regular activity and milestones:
    - Complete at least two stories
        - First one focus on *process* for a simple feature or fix
            - Create and task out work
            - Run company software. Debug. Run solution.
            - Write automated tests in the solution
            - Manage source code in the solution
            - Create a Pull Request for a small feature or fix
            - Complete a code review in the solution
        - Second one *increase the programming challenge* as level-appropriate possibly with a database and/or in a hosted environment with queuing technology
            - Create and task out work
            - Run company software. Debug.
            - Write automated tests in the solution
            - Manage source code in the solution
            - Create a Pull Request for a small feature or fix
            - Complete a code review in the solution
    - Attend stand-ups and other team meetings as available
    - Work on the individual professional development plan and update it for after the program