Java Bootcamp: Object-Oriented Programming

Course Description

In this course, apprentices will learn object-oriented programming using the Java programming language. The course will cover the fundamentals of the Java language, file I/O, the MVC design pattern, dependency injection, unit testing, and working with outside libraries using Maven. By the end of the course, apprentices will be able to develop complex and robust console applications.

Course Objectives

Upon successful completion of this course, apprentices will be able to:

1. Use Java to create a program that meets a client's specifications.
2. Demonstrate appropriate use of code layout, object names, and comments to make code more understandable to other developers.
3. Demonstrate the use of Maven to manage outside libraries used in a project.
4. Demonstrate the use of dependency injection in the construction of an application.
5. Build applications using the MVC pattern.
6. Incorporate I/O operations in an application to store and retrieve data used by the application.
7. Demonstrated appropriate data type use when developing an application as well as the ability to convert data types when necessary.
8. Demonstrate appropriate use of data types for stored values.
9. Create an application that has a service layer with business logic
10. Create an application that uses lambdas and streams in at least part of the DAO.
11. Incorporate unit tests for DAOs and the service layer in an application.
12. Demonstrate use of a List or Map to store DAO data in memory.
13. Create an application that validates user input.
14. Create an application that properly handles all exceptions.
15. Explain the foundational elements of the Java language, object-oriented design, and testing methodology.
16. Explain dependency injection, the advantages of loosely coupled code, and the MVC pattern.
17. Explain the importance of the Service Layer in application architecture.

Course Technology

Each apprentice must provide their own computer with a current operating system (Windows, Mac OS, or Linux) and the apprentice must have administration rights to install and updated software on that computer.

In addition, the course will use the following technology:

- A learning management system (LMS) that houses all instructional content.
    - Apprentices will receive information about how to log into the LMS prior to the start of their first course.
    - Apprentices will be enrolled automatically into required courses and receive notification by email of such enrollment.

- An integrated development environment (IDE):
    - Instructions for installing the required IDE are provided in each course.
- GitHub to manage and share code
- Slack, for interaction with instructors and other apprentices

Individual courses may include additional software requirements, as described in the course.

All required software is either free or provided at no cost to apprentices.

Course Prerequisites

Apprentices are expected to have completed the following Software Guild courses:

- How to Guild
- Introduction to Web Development

Methods of Instruction

The Software Guild uses a master-apprentice approach to learning. This means that most learning is hands-on, where the apprentice will complete tasks assigned by the instructor on their own or in collaboration with other apprentices, and the instructor is available to guide apprentices toward learning how to complete those tasks.

This course includes the following methods of instruction:

- Content presentation using text and videos
- Hands-on activities, including code-alongs that walk apprentices through a solution and exercises that apprentices will solve without guided instructions
- Quizzes
- Code assessments

The classroom version of this course uses a flipped course approach to instruction. In this approach, apprentices will review the instructional content in the LMS outside of class time, as assigned by the instructor, and come to class prepared to use that material. Class time is then reserved for asking questions about that material and applying the material to a real coding solution presented by the instructor.

Equivalent coding exercises are provided in the online version of each course so that even online learners can take advantage of this approach to learning.

All apprentices should thoroughly review the content for each topic in the online course and maintain an engineering notebook as a record of their progress. All apprentices (classroom and online) are expected to use Slack to ask questions and collaborate with other apprentices outside of the classroom or LMS.

Course Policies

The course polices listed below are summarized from the Apprentice Contract all apprentices are required to sign. See the Apprentice Contract for details about all course policies.

Academic Honesty

Academic honesty is expected at all times. Scholastic dishonesty includes, but is not limited to: copying, cheating, plagiarism, obstruction, collusion, misconduct, computer misuse, and unethical conduct. Appropriate consequences will apply to each situation. Any first instance of academic dishonesty will result in a warning; a second instance will result in expulsion.

Attendance

Classroom learners are expected to attend each scheduled class, and excessive absences may result in dismissal. While in class, apprentices should focus on the task at hand and treat the equipment and physical surroundings as a professional workplace setting.

Online learners are expected to spend a minimum of twenty hours per week on the course material and to attend scheduled live sessions with the instructor.

Special Accommodations

In the event that an apprentice has an identified disability that may interfere with the learning process, the apprentice should contact their Operations Manager for assistance with appropriate accommodations.

Assessment Activities

The course includes several types of assessments.

- **Quizzes** are graded automatically within the LMS, but the grade for most quizzes does not count toward completion of the course. Apprentices may retake any quiz as many times as they wish to improve their score.
- **Coding Exercises** are provided to give apprentices the opportunity to solve problems on their own (in class or online), without the step-by-step instructions included in a code-along or lesson. These activities are not graded, but they do provide apprentices with the chance to see how well they can apply the concepts included in the lessons and code-along. Apprentices are expected to ask for help if they run into problems completing any exercise.
- **Code Assessments** are formal assignments that must be submitted to an instructor for review at the end of each milestone and at the end of each course. These activities typically provide only a basic overview of the problem to be solved, with the goal of allowing apprentices to solve the problem using skills and concepts presented in the course up to that point.
- A **Code Mastery Assessment** must be completed at the end of the final course in the program. This assessment determines completion of the program.

Code Assessment Policies

Software development is a hands-on activity that requires both an understanding of the concepts that underlie the code and how to use code appropriately. Unlike other areas of study, there are typically multiple ways to reach an expected goal, and apprentices will be evaluated as much on the steps they used to reach the goal as on the goal itself.

The following policies apply to the code assessments:

- Upon completion of a code assessment, the apprentice will follow the instructions provided in the course to submit their work to Crucible and submit a link to their Crucible assignment to the assignment page for the code assessment.
- The instructor will schedule a time to meet with the apprentice to perform a code review, where the instructor evaluates the code and interviews the apprentice about the project.
- The instructor will assign one of three grades to the code assessment:
    - **Meets Expectations:** Submitted work meets technical requirements and the apprentice demonstrates sufficient knowledge of supporting topics.
    - **Needs Improvement:** The submitted work does not meet technical requirements or the apprentice cannot demonstrate sufficient understanding of the supporting topics.
    - **No Submission:** Each apprentice is expected to turn in whatever work they have by the posted deadlines, even if the work is not completed. No credit is received for work that is not submitted.

In the event that an apprentice receives a grade of Needs Improvement:

- It is up to the apprentice to manage their time appropriately to fix any deficiencies.
- It is up to the apprentice to seek additional help from staff as needed.
- It is up to the apprentice to schedule time with the instructor to review the updated work or understanding.

The grade will be changed to Meets Expectations when the apprentice completes the work and successfully presents it to the instructor.

Each code assessment includes a rubric that identifies the topics the instructor will include in the code review. Apprentices should review the rubric prior to completing the code review.

Communication and Professionalism

Communication and professionalism are evaluated throughout each course. Apprenticeship is like a job, and like a job, an apprentice can be dismissed for not acting professionally. All apprentices are expected to:

- Communicate early and often about technical and personal struggles.
- Complete assignments following all directions.
- Manage their time well to allow enough time for assignments and getting help before the day they are due.
- Be respectful of other apprentices, staff, and guests, in person and online.
- Check email and Slack regularly and respond to communications from staff promptly.

Classroom learners are expected to:

- Show up on time and ready to work each day.
- Communicate early about any absences or tardiness with instructors.
- Use common sense dress code.

Online learners are expected to:

- Log into the course regularly and spend at least 20 hours a week working through the course material.
- Communicate early about any issues that may prevent the apprentice from keeping up with the course.
- Respect instructor schedules of availability.

Instructors monitor apprentice behavior throughout each course and keep records on attendance, the ability to meet deadlines, and communication between the apprentice and the instructor, as well as between apprentices. These records are used to determine whether or not each apprentice has met expectations at the end of the program.

Course Completion and Deadlines

An apprentice successfully completes the course by receiving a Meets Expectations on all code assessments for the course. The assessments for all milestones must be completed within five weeks of the start date, and the final assessment must be completed within six weeks of the start date.

An apprentice cannot continue on to the DDWA portion of the program without completing the OOP portion of the course.

Program Completion

In order to receive a certificate of completion for the program, an apprentice must:

- Complete the OOP course with a Meets Expectations on each code assessment.
- Complete the DDWA course with a Meets Expectations on each code assessment.
- Receive a Meets Expectations for both areas of Communication and Professionalism, based on conduct throughout both courses.
- Receive a Meets Expectations in two of the three areas of software development evaluated in the Mastery Assessment at the end of DDWA, including database programming, server-side programming, and front-end programming.

Disciplinary Actions

There is normally an escalation procedure if an apprentice violates expectations:

- **Verbal Warning** – An instructor and another Guild employee have a conversation with the apprentice, and they document the conversation in the apprentice's records.
- **Written Warnin**g – The apprentice must sign the documentation and the documentation is placed in the apprentice's records.
- **Dismissal** after two warnings for the same issue (including professionalism).

Depending on the severity of the infraction we may move directly to dismissal.

Failing the Course

Our success in placing our graduates relies on employers trusting us to provide them with valuable employees.

If we pass people who are not ready, we are doing a disservice to them, the employers, and future apprentices (who may miss out on jobs from dissatisfied employers).

If an apprentice does not complete the assessments in OOP with a Meets Expectations grade by deadlines outlined above, they cannot continue into DDWA. In that case, the apprentice may discuss the viability of the following options with an Operations Manager and instructor:

- Move to a future in-person cohort
- Move to online
- Withdraw

If an apprentice does not complete the assessments in DDWA with a Meets Expectations grade by the stated deadlines, they will not receive a certificate of completion for the course, and they will not be eligible to take advantage of our employer network.

Content Outline

Milestone 1: Java Basics

Milestone Description

In this milestone, we will start building a strong foundation in Java language fundamentals. We will learn how to use Java data types, keywords, expressions, and operators that will let us build more complicated applications. We will also learn about the tools we use to more easily develop these applications.

Milestone Objectives

Upon successful completion of this milestone, apprentices will be able to:

- Create applications in Java with appropriate types and variables.
- Demonstrate the use of Java expressions and operators.
- Demonstrate appropriate naming conventions for different parts of an application.
- Demonstrate appropriate use of code layout, object names, and comments to increase code readability.
- Use conditional statements and loops to control the flow of a program.
- Use methods to organize code into re-usable blocks.
- Use arrays to store collections of data elements.
- Use IDE tools to debug application code.
- Use a flowchart to describe the flow of a program.

Milestone 2: Classes and Objects

Milestone Description

In this milestone, we will begin to learn how to better organize our code. We will take advantage of objects and the layers of the MVC design pattern to build our program. We will also incorporate the ability to store collections of data in memory or in a file. Finally, we will look at the Agile development methodology and how we can apply that to the programs we are creating.

Milestone Objectives

Upon successful completion of this milestone, apprentices will be able to:

- Identify basic principles of object-oriented programming.
- Create custom classes in a program.
- Use dependency injection in a program and understand the benefits of loosely coupled code.
- Organize code by responsibility into types and layers using the MVC pattern.
- Explain the use of interfaces, composition, and inheritance in a program.
- Use more than one object in a program.
- Marshal and unmarshal data.
- Incorporate I/O operations in a program to store and retrieve data.
- Demonstrate the use of a list or map to hold data in memory.
- Explain the Agile development approach to building software.

Milestone 3: Intermediate Java

Milestone Description

In this milestone, we will add a service layer to an application and demonstrate the use of unit tests to identify problems in the code. We will also look at how to incorporate enums, the Java DateTime API, lambdas, streams, and aggregate functions into an application to make it more robust.

Milestone Objectives

Upon successful completion of this milestone, apprentices will be able to:

- Demonstrate the use of the service layer in an application.
- Perform unit testing using the JUnit framework.
- Incorporate enums into an application.
- Demonstrate appropriate use of the Java DateTime API.
- Demonstrate appropriate use of BigDecimal in an application.
- Incorporate lambdas, streams, and aggregate functions into an application.

Milestone 4: Advanced Java

Milestone Description

In this milestone we will finish up with a look at the powerful Spring framework. Apprentices will see how to use Spring for dependency injection and take a closer look at the Maven build tool.

Milestone Objectives

Upon successful completion of this milestone, apprentices will be able to:

- Describe and demonstrate the functions of Maven as a programming tool.
- Explain the Maven lifecycle.
- Explain how Spring fits into Java development.
- Use Spring for dependency injection.
- Explain how dependency injection and programming to interfaces work together.