

## COURSE OVERVIEW

**Course Title:** Level 0: Loops & Logic

**Course Number:**

**Number of Units:** 1.5 units

**Total Hours of Instruction:** 36

### Course Description

During this course, students acquire basic programming skills in Java through practice. Using professional programming tools such as Eclipse and GitHub, students end this course with mastery of logic, loops and variables, and with a firm foundation in problem solving and logical thinking. In order to make this fun and interesting for kids, we use loops and logic to make simple games, user interactions and animations. The challenge-based curriculum encourages a *growth mindset* that requires building the student's confidence in themselves as problem solvers. This course is designed for beginning programmers aged 9-15 years old.

### Content and Evaluation

The curriculum for this course is a proprietary set of "recipes" that use the intentional method to guide students in creating their first Java programs. A recipe is a set of instructions that students convert into code giving them a thorough understanding of the effect of each line of code they write. The recipes gradually increase in granularity until the student can write programs by themselves from scratch.

Among other tools, we use an in-house library, derived from Logo's Turtle that allows the students to move a robot around the screen by writing Java. We also write user-interactive programs and games using Eclipse, and create code-based animations with Processing.

The 1.5 hour exam that completes this course includes a written portion, and two coding exercises that must be completed without external help.

### Extra Credit

n/a

### Methods of Instruction

- |                                     |                                    |                                     |                                |
|-------------------------------------|------------------------------------|-------------------------------------|--------------------------------|
| <input checked="" type="checkbox"/> | CLASS DISCUSSION/DISCUSSION BOARDS | <input checked="" type="checkbox"/> | LECTURES                       |
| <input type="checkbox"/>            | FIELD TRIPS                        | <input type="checkbox"/>            | CASE STUDIES                   |
| <input checked="" type="checkbox"/> | GROUP WORK                         | <input checked="" type="checkbox"/> | OTHER: PROGRAMMING ASSIGNMENTS |

### Out of Class Assignments

Total hours expected to complete assignments: n/a

- |                          |                    |                          |   |
|--------------------------|--------------------|--------------------------|---|
| <input type="checkbox"/> | TEXTBOOK EXERCISES | <input type="checkbox"/> | READINGS  |
| <input type="checkbox"/> | GROUP WORK         | <input type="checkbox"/> | WRITTEN ASSIGNMENT/ESSAY(S)                           |
| <input type="checkbox"/> | STUDENT PROJECT    | X                        | OTHER: EXPLORATION OF CONCEPTS BY PROGRAMMING AT HOME |

#### Evaluation/ Grading

- |                          |                                  |                          |                                       |
|--------------------------|----------------------------------|--------------------------|---------------------------------------|
| X                        | EXAM(S)                          | <input type="checkbox"/> | CLASS PARTICIPATION/DISCUSSION BOARDS |
| <input type="checkbox"/> | WRITTEN ASSIGNMENT/ESSAY(S)      | <input type="checkbox"/> | CLASS PROJECT(S)                      |
| X                        | OTHER: TWO PROGRAMMING EXERCISES |                          |                                       |

### Topical Outline

Rather than follow a traditional lecture-style/textbook form of teaching, this course is driven by a set of practical learning objectives that are practiced until the student gains mastery over that skill. In the same way a shoemaker might make a lot of shoes before becoming a master, our students continually write code to solve a particular challenge in the theme of the skill they are learning. As a shoemaker would not learn to make shoes by spending time listening to someone describe the theory of lasting and hammering, a young programmer does not learn by listening to an adult explain the theory of computer science.

At the end of Level 0, students have mastered these skills;

- How to write and run Java code within the Eclipse IDE.
- How to create a class and start it running with the `main()` method.
- How to print something to the console using `System.out`.
- How to implement logic using `if`, `else if` and `else` statements.
- Basic and nested `for` loops.
- Use of `JOptionPane.showMessageDialog()` and `JOptionPane.showInputDialog()` to interact with the user.
- How to make code-based drawings and animations using Processing.
- How to concatenate strings and variables.
- How to declare and initialize `String` and `int` variables.
- How to compare `int` and `String` types using `==` and `equals()`.
- The logical operators `!=`, `==`, `<=`, `>=`, `<` and `>`.
- How to call methods.
- How to get a random number between two values.
- How to convert a `String` to an `int`.
- How to read code and spot errors in it without an IDE.
- How to download and publish code on GitHub.
- Put these skills to creative use.