



The LEAGUE

of Amazing Programmers

Igniting young minds through computer programming



igniting young minds

Writing computer programs is a challenging, yet extremely satisfying personal experience that develops essential skills in logic, problem solving, collaboration and creativity. It is uniquely suited to train children to adapt attitudes of confidence and endurance when faced with any challenge. Coding promotes a way of thinking that can help children in every area of life; whether seeking to understand how complicated systems like economies work or solving a problem in a stepwise fashion.

OUR APPROACH

Our students code on their very first day at The League. This hands-on approach engages students and allows them to learn by doing. As opposed to teaching methods with a heavy emphasis on theory, our students learn how to code by coding, and theory is introduced as they code. This practical approach lends itself to a collaborative learning environment; as students actively learn how to program, they share their code with one another and help each other troubleshoot, which creates a sense of teamwork and community in the classroom.

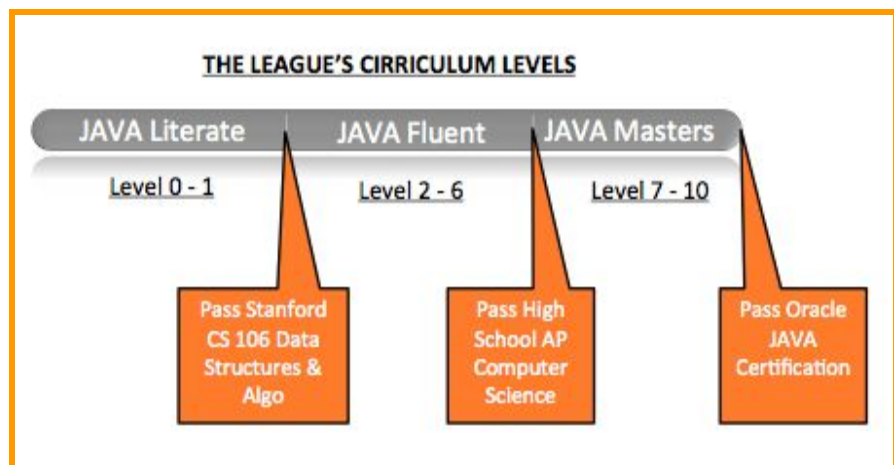
Our curriculum emphasizes fun and discovery. Each lesson contains high-interest content for kids, such as video games, social media, mobile apps, and robot automation. We use proprietary “recipes” which are step-by-step instructions that give students a thorough understanding of the effect of each line of code until they are capable of writing it from scratch. Each recipe has a learning goal, but the students are usually too excited about the end result to realize that they are learning complex programming concepts. Lessons evolve as new technologies appear, and we improve them iteratively as we find what works best for stimulating young minds.

Our core teaching values heavily influence our curriculum. In addition to teaching superior coding skills, we aim to develop confidence and a growth mindset in our students. We hope these characteristics will better equip students to overcome any problem, in programming and in life, by breaking it down and tackling its smaller parts. Most of all, we want students to have an association of *joy* with coding, which is why we try to make learning at The League as fun as possible.

CURRICULUM AND LEARNING STRUCTURE

Overview

Students are taught in-person by experienced software engineers. Students attend one 90-minute class each week and progress through ten curriculum levels at their own pace by passing an exam at each level. Once they have completed Level 8 (Oracle Certification), our students are fully equipped and ready for the workplace. We also use external entities to certify our students; materials from Stanford CS106 in Level 3, and the AP Computer Science exam in Level 5.



10-Level Curriculum

Every level of our curriculum is accredited for college prep credit through the UCSD Extension Program.

Level 0: Loops & Logic | 4-6 months

Basic input/output, "for" loops, boolean logic (if/else)

In this level, students acquire basic programming skills in Java. 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. Students produce simple games, user interactions and animations. The challenge-based curriculum encourages a *growth mindset* that stimulates curiosity and the joy of discovery and learning. It is also essential to building the student's confidence in themselves as problem solvers.

Level 1: Object-Oriented Programming | 9-12 months

Swing GUIs, Objects, Methods

We learn Java's Swing libraries which give students the ability to create elegant Graphical User Interfaces (GUIs) from scratch, while becoming familiar with the patterns and paradigms of object-oriented programming (OOP). Understanding OOP requires abstract thinking about technical concepts, but mastering this mindset lends itself to breaking complex problems into more manageable ones, which is a paramount skill for any software engineer. OOP is a high-level concept which lays the foundation for writing code in a sharable and accessible way.

Students write a substantial amount of code during this level, which is published online as part of their growing GitHub portfolio.

Level 2: Build Your Own Game | 4-6 months

Design, develop and implement a game

At this point in our curriculum, our students have gained some programming chops, so it's time to put their skills to work. We try to limit the focus on new concepts, and mentors act as a guide in the creation of a game that reflects the student's own interests and style. Taking an imaginary idea and developing it into functional software is a landmark moment for any new programmer.

Students complete this level with a fully usable game that includes multiple-iterations of usability testing, publishing, and a formal demo. This level emphasizes UX (user experience) testing so that students start thinking about testing and the user's perspective. The projects are publicly published on GitHub and form part of the student's professional portfolio of work.

Level 3: Data Structures & Algorithms | 4-10 months

ArrayLists, HashMaps, basic algorithms (sorting, searching), JUnit, Test-Driven Development

We begin to study college-level material at Level 3 and place heavy emphasis on test-driven development (TDD). Students learn to master common data structures such as lists, stacks & hashmaps and to implement basic sorting and searching algorithms. It's the beginning of a long and beautiful relationship with unit testing and JUnit. These developing skillsets enable students to write more stateful and complex games which are tested on maze-navigating robots.

League students test out of this level using a mid-term exam for first year students of computer science at Stanford University. By that time, their practical coding skills are on a par, if not beyond, students at upper-division computer science college students.

Level 4: Java Ninja | 6-12 months

Inheritance, Polymorphism, Casting, Composition, Arrays, File I/O, Exceptions, Lambdas, Streams

Upon completion of this level, students have a solid understanding of Java and are internship-ready. They have complete knowledge of the fundamental features of programming and have practiced that knowledge enough to work fluidly with code; it feels different, like how driving a stick-shift compares between your first week and your second year. At this point they are able to code at the level of a junior developer.

Level 5: AP Computer Science | 2-3 months

Before entering level 5, students already know how to practically apply all the concepts on the AP Computer Science Exam. In Level 5, students learn how to write code on paper and learn theoretical Computer Science topics. Given the amount of code our students have already written by this point, we are mostly studying exam-taking in this level, and passing the AP test is a breeze.

Level 6: Project for Social Good | 6-8 months

The skill of programming is a powerful tool that can be used to change the world. That's why in Level 6 our students team up and spend 6-8 months working on a project that impacts society for the good. These are usually mobile or web apps and involve databases or web services. Some of our students have won prizes competing against adult teams at civic hackathons with these apps.

Level 7: Software Craftsmanship | 6 months

Unit testing & TDD, Design patterns, Code smells & refactoring, Clean code

In this level, students hone their craft by studying software architecture and industry best practices. This course has been carefully developed in collaboration with some of the best and most experienced programmers in San Diego. This stuff isn't even taught in college!

Level 8: Oracle Certification | 4-5 months

Oracle Java SE 8 Programmer I, 1Z0-808

In this level, students learn the intricacies of the Java programming language. The Oracle Certification exam tests you on all the dark corners of the language, and it is an exam that even professionals must study for.

Level 9: Ready To Work

Students have now completed all League coursework and are prepared to take on an advanced degree



or start their career. As they prepare for these big life transitions, the majority of them volunteer extra time to The League helping as technical interns and TAs since by this point they are practically family.

OTHER LEAGUE LEARNING OPPORTUNITIES

International Autonomous Robot Competition (iAROC)

For the last 8 years, The League has hosted the Autonomous Robotics Competition (iAROC), which will take place at the San Diego Central Library this year. Students form teams and work together to program their own robots to compete in a series of challenges in the form of mazes and games. This competition emphasises robot programming over robot building, providing an opportunity for students to see the interconnectivity between software and hardware and to showcase their learning on a larger scale. The League provides scholarships to qualifying teams in addition to technical support and mentorship for all participants, including a week-long RoboCamp where teams can prepare and practice with sample mazes. See the website <http://iaroc.org/> for current details.

Summer & Holiday Workshops

Occasionally we offer specialty workshops like Build Your Own Bot, Unity Gaming, etc. for special interests and learning enhancement during the summer and holiday vacation periods.

Teaching Assistants

Our older students have opportunities to volunteer as Teaching Assistants during the week-long Intro to Java and other workshops as well as in the lower-level weekly, ongoing Java classes. Mentoring other students is a great way to build confidence, affirm their personal growth, and deepen their understanding of their own learning through teaching it to others.

Jobs & Internships

We connect our students with leading technology companies throughout San Diego for job and internship opportunities during their high school years and thereafter. Our high school students are able to code from scratch, at a level equivalent to a junior developer with a 4-year Computer Science degree. In addition, The League offers several internal internships for programming, web design, teaching, graphic design, grant writing, fundraising, and market research.

