edhesive

# Explorations in Coding: Introduction to Python 1

In this introductory course, students will learn foundational concepts and skills of programming and computer science (CS). The course is designed to be fun, engaging, relevant, collaborative and accessible for all students, regardless of background or prior experience. Explorations in Coding students will build their understanding of programming and computer science through interactive coding explorations, practices, and projects in Python, where they will create animations, graphic designs, and other interactive programs. Supplemental online and unplugged activities foster students' deeper understanding and allow them to demonstrative their creativity, communication and critical thinking skills. Students are engaged and motivated as they quickly learn how computer science impacts the world around them and relates to their own interests and experiences.

## Prerequisites

No prior knowledge or experience is necessary for this course.

## Grade Levels

Appropriate for middle school or early high school grades.

## Course Goals

- Introduce computer science as an engaging and relevant discipline.
- Develop foundational skills and knowledge in programming and computer science.
- Strengthen problem solving and critical thinking skills.
- Foster creativity, collaboration, and communication.
- Explore issues raised by present and future societal impacts of computing.
- Demonstrate that all students can be successful in computer science.

## Curriculum Overview

This course is organized into nine modules. Each module builds computer science competency through highly engaging, interactive coding projects as well as a series of online and unplugged activities that introduce students to foundational computer science concepts and skills, reinforce core programming concepts, and support students in strategic problem solving practices.

### Course Modules

# Module Overviews

## Module 1: Introduction to Computer Science

Students will explore how computer science is connected to their own lives and how computing helps solve big, real-world problems. Students will view themselves as creators of technology as they begin coding in Python and create their first programs. Topics include:

- What is computer science?
- Programs and programming
- Hardware and software
- Computer science and me

## Module 2: Coding in Digital Space

Students are introduced to the problem solving process and how it can be used to solve real world and coding problems, a practice integral to computer science. Students will use this process to create programmatic animations and drawings as well as to debug those programs. Topics include:

- Problem Solving Process
- Measurements in Digital Space
- Debugging
- Inputs and Outputs

## Module 3: Shapes and Drawing

Students will learn about variables, parameters, and comments through a series of cooperative learning and coding activities centered around graphic design problem solving. Students will hone their use of the problem solving process by examining the idea of establishing program requirements. Topics include:

- Variables and variable names
- Modifying parameters
- Writing comments
- Program requirements

## Module 4: Intro to Logic

Students are introduced to two of the three core building blocks of computer programs - conditionals and loops. Students will combine the use of these logic structures with user input to create programs like chatbots that are interactive or animations that are more dynamic. Topics include:

- Conditionals
- Loops
- User input
- Brainstorming solutions

## Module 5: Putting Things in Order

Students will consider sequential operations, the third of the core building blocks of computer programs. Students practice with sequence as they order real world and programmatic algorithms with pseudocode or code. Students will also learn about the final steps to the problem solving process, testing and iterating. Topics include:

- Ordering algorithms
- Pseudocode
- Testing
- Iteration

## Module 6: Exploring Variables – Part 1

Students advance their coding knowledge as they work with various data and variable types, such as strings and random integer variables. Students extend their learning about computer science through lessons on diversity in computer scientists and cryptography. Topics include:

- Data and variable types
- Random number generators
- Diversity in computer science
- Cryptography

## Module 7: Exploring Variables – Part 2

Students build upon their coding knowledge by integrating new variable types into simulation-based programming activities. Students will learn new ways to use variables to control more than one object or output at once. Topics include:

- Multiple variable types
- Variable uses
- Combining strings and string variables
- Computer simulations

## Module 8: Math and Computation

Students will learn to write programs that solve practical mathematical calculations. Students explore the history of computation by learning about and contrasting human computers to electronic computers. Topics include:

- Data and variable types
- Random number generators
- Diversity in computer science
- Cryptography

## Module 9: Cumulative Review

Students demonstrate the ability to use and combine computer science and coding concepts from throughout the course. Topics include:

- Cumulative review through coding practices
- Cumulative project
- Cumulative test

## Course Materials

All core course materials and resources are provided for this class. Additional materials may include:

- Screen-cast or projection device to project provided lesson slides or videos
- Printed copies of provided activity sheets and materials
- Paper, pens/pencils, and other basic classroom supplies

*Optional*:

- Pocketed folder or binder
- Headphones/earbuds (for watching instructional videos)

## Differentiation

The Explorations in Coding courses provide adaptive scaffolding, indicators of common misconceptions, and diverse activity types to provide a dynamic curriculum strategically designed to support students with varying academic backgrounds or previous experience with technology. Coding extensions and other challenge activities are additionally provided throughout the course materials to engage and motivate advanced or accelerated students.

## Standards-aligned

The Explorations in Coding courses are aligned with the Computer Science Teachers Association (CSTA) K-12 Computer Science Standards and the Common Core State Standards (CCSS): Standards for Mathematical Practices. The CSTA K-12 Computer Science Standards are the national benchmark for introducing fundamental concepts in computer science at all grade levels. The CCSS Standards for Mathematical Practice outline key processes and proficiencies for effective problem solving. The Explorations in Coding lesson guides indicate each module’s alignment to both sets of standards.

## About Explorations in Coding

Computer science is the science behind computing and learning how to use the power of computers to solve big problems. It is the study of the ideas, ways of thinking, programming languages, hardware and software needed to solve problems with computers, and encompasses the study of computers and algorithmic processes, their principles, their designs, their applications, and their impact on society.

The Explorations in Coding courses seek to reinforce the understanding that computer science is intellectually important and that all students should have a basic understanding of computer science and programming principles, technology enabled applications from calculators to simulations, and awareness of the societal impacts of computing. Some key values for advancing computer science education for students include:

**Computer science leads to multiple career paths**. There are tremendous job opportunities in computer science. Professionals in every 21st century discipline need to understand computing to be productive and competitive in their fields.

**Computer science teaches problem solving**. Computer science requires students to apply critical thinking skills to consider the context of the situation, work cooperatively on solutions, and balance the robustness, user-friendliness, and efficiency of solutions to real-world issues.

**Computer science supports and links to other disciplines**. Computer science is not simply one more discipline to learn; rather, it is a methodology that enables the study of and innovation in other disciplines.

**Computer science can engage all students**. It allows hands-on practice and offers students opportunities to solve computational problems relevant to their own interests, passions, and experiences.

## Essential CS Practices

The Explorations in Coding courses seeks to reinforce the following essential computer science practices:

- Problem-solve: find solutions to challenging, real-world problems
- Persevere: try again and again, even when something is very hard
- Collaborate: work together to achieve something greater than could be done alone
- Create: design and develop interesting, relevant computational artifacts
- Communicate: strengthen written and verbal skIlls to describe computing and one’s work
- Think critically: identify impacts of computing; draw connections to the real world

## Student Collaboration

This course is designed to allow teachers to encourage and support student cooperation and collaboration. Students will work on many of the coding and unplugged activities in pairs, groups or teams. Even if students do not choose their own partners, they should be responsible for working well together. For examples, if using pair programming, students should frequently trade roles as “driver” (controlling the computer) and “navigator” (assisting the driver and keeping him/her focused on the big picture), and students should view their teammates as sources of primary support during those activities. To encourage this, students can be encouraged to ask their partners and teammates for help before asking the teacher.

## Student Guidelines

The following are sample guidelines for student expectations for the course. These may be modified or adapted by teachers. Many aspects of this course will be exciting and extremely rewarding, though some parts will not be easy. Those parts may be very challenging and even frustrating at times, and this is okay. We consider mistakes and failure as our first attempts in learning, and we persevere and learn from these. In this class, teachers and students work and learn together to achieve more than they could do alone.

Students are expected to follow all school and classroom prescribed rules. Some sample classroom rules can include:

- Follow teachers’ directions.
- Complete all assigned work (to the best of one’s ability).
- Come to class prepared and on time.
- Keep the classroom clean and orderly.
- Use computers and the Internet responsibly.

Sample classroom norms can include:

- Be respectful.
- Be safe and responsible.
- It’s okay to make mistakes.
- It’s okay to not have the answers (yet).
- Never give up. (And, don’t let your teammates give up.)

## Syllabus Note

This syllabus is subject to change; and should be considered a template that teachers may use and adapt to their classrooms.