Intro to Coding – Teachers' Guide

overview

BACKGROUND

TEACHING

LEARNING

BACKGROUND

What This Class Aims to Be

This course was made to make it easier for educators and students to experience the computer science and information technology fields. With new education and technology concepts in mind, the Intro to Coding course was curated with the goal of introducing more schools into programming classes through a modern, free, and simple path.

Who This Class Is For

This course is for young learners with an interest in computers and coding. This course can be used in middle schools and high schools recommended for students age 12 to 18. Although this course was made for kids with differing levels of understanding, we recommend that students are not required to take this course.

Programming in the Real World

This class hopes to give a look into what programmers in the real world do every day. Education and programming are two objects that don't always go well together. Today, technology changes at an unprecedented rate, and education can't always keep up to teach what's new. As a result, many approaches to teaching programming cover the fundamentals and principles that are unchanging. This is a route that Collegeboard's Advanced Placement Computer Science courses take. However, taking this approach discourages numerous students who have a natural interest in these topics but find learning about them uninteresting and tiresome. This course intends to balance a level of new technology with basic concepts that keeps learning about programming interesting and useful at the same time.

How This Course Was Made

This course was made in collaboration with new teaching and new programming ideas. A main goal of this course is to utilize present-day learning methods to teach present-day material. We wanted to update obsolete programming classes with new and effective learning, in the easy to adopt form. We chose to teach this course in Python, a simple-to-learn modern high-level language.

What You Can Do with This Course

This course is completely free and open source. You are free to modify any and all parts in order to better suit your classroom and redistribute your additions for other educators to use. We ask that you do not sell any of these course materials and further the

community with your contributions. Any changes you would like to see made to the source material can be submitted for review on our GitHub page, as an issue.

TEACHING

Teaching Requirements

In order to properly meet student's needs and allow for a successful learning environment, the teacher of the Intro to Coding course must be able to []. This course is introductory and aims to bring teachers and students alike into the Computer Science field, with little extraneous effort.

After preparing to teach this course,

You will need:

- programming knowledge
- fluency with the Python language
- complete understanding of all covered topics
- individual time with each student

It's beneficial to have:

- experience with other programming languages
- "real world" programming experience
- computer/technology background

You do not need:

- a degree in computer science or related field
- many years of experience

Preparing to Teach This Class

Preparing to teach this class will take varying levels of effort depending on one's prior experience. The following is a guide on what to do to prepare to teach this course.

Programming Experience	Preparation	Timeframe
Expert	Review course videos, labs, and projects.	2 weeks
Intermediate	Review course videos, labs, and projects, complete some	
Beginning		

None	Work through course material, complete all labs, projects, and quizzes	

Class Length

This course is not a full year curriculum. This class can be taught as one semester or one trimester. For students who would like to continue learning programming, we recommend the Intro to Coding course be taught alongside the Intro to Programming course. Intro to Programming teaches more advanced programming concepts built off of the basic tools learned in Intro to Coding. The two courses together can be used for a full year or two trimester schedule.

Classroom Requirements

To take full advantage of this course's learning system, your classroom will require the following:

- personal computers
    - access to file system with basic management rights
    - VS Code, Processing, Python, and other used programs installed
    - Compatible with Windows, Mac OS, or Linux
- a reliable internet connection
- personal headphones or speakers (not recommended)

LEARNING

Material

The primary learning material for students is a set of videos explaining each topic in detail with general examples. We believe these videos enough for the average student to build a strong understanding of assigned and tested topics. However, deeper understanding of each topic is only attained through self-experimentation, experience, and questions. No extra costly materials are required to teach Intro to Coding. All videos are available for download and publicly accessible on YouTube. Optionally, there are text transcripts for each video inside of the 'resources' folder inside the course materials.

Resources

Many other resources for students can be found on the web. Useful links and books are listed in the course materials, but not required.

Difficulty

This course aims to provide a healthy and reasonable challenge to students, while providing space for other classes. Intro to Coding is best suited as an elective pace course.

Pacing

This course is a flexible self-paced course where students are able to take time on topics they struggle with, and quickly move through ones that come easily. We encourage a soft deadline for all labs, projects, and quizzes, as an indicator of where students should fall. Penalization for late work is recommended for students who fall behind more than two weeks. A recommended schedule for semester and trimester schools follows:

Semester Schedule	Trimester Schedule
Week 1 – 1.1, 1.2	
Week 2 – 1.3, 2.1	
Week 3 – 2.2, 2.3	

Testing

In any programming class, understanding and skill may not always translate onto paper. Deeper understanding of programming is most apparent through project work, so we encourage classrooms to the use unit projects as a means of assessment grades. Unit tests are not provided. Lesson quizzes are provided and optional to classrooms.

Answer Keys

The answer keys for labs, projects, and quizzes cannot be found online. To restrain students from cheating, we ask that you send an email to [] with your name and school name for the answer keys, and we ask that you keep from uploading them yourself.

Grading

This class adopts standards-based grading on a four-point scale. Each assignment uses standards to grade the students' understanding of the material. The standards are graded as following. Grading scales may be adjusted for each school.

4 – exemplary	A - 100-90%
3 – accomplished	B - 89-80%
2 – developing	C - 79-70%
1 – beginning	D - 69-60%
0 – bad	E - 59-0%

FURTHER QUESTIONS

If you have any questions about the Intro to Coding course, questions about teaching this class, or any other feedback, please let us know! Email us at [] and we'll try our best to address your questions.