

e2020 Curriculum Briefing Computer Science I (EL5722)

Course Description

In this introductory computer science course sequence, students will learn the basics of programming through hands-on projects that encourage creativity and experimentation. Students will apply key computer science concepts to programs created in and run by Python, a programming language with simple syntax and flexible, robust performance. Python runs on most operating systems and is used professionally to write web applications, desktop applications, and many other types of software programs. Students will learn about the real-world experiences of programmers through personal questionnaires with professionals in the field of computer science. By the end of this two-semester course sequence, students will have many of the fundamental skills and conceptual background needed to get started as a programmer in Python or in any programming language. Project-based learning walks students through writing code, while course assignments build on what students learn.

This course is aligned with the Washington State CTE Computer Programming framework and the Texas State TEKS Computer Programming framework.

Topics of Study

- Starting with Python
- Creating a graphical Hello World program
- Variables
- Loops
- Creating a drawing program
- Software development
- Strings and lists

e2020 Curriculum Briefing (continued)

Computer Science I (EL5722)

Course Features

- While explaining concepts and code syntax to support student learning, the course provides step-by-step instruction on how to write code.
- The course uses Python's built-in, integrated development environment, IDLE, for writing code. IDLE provides a command line programming environment for testing individual lines of code, as well as a text editor for writing longer programs.
- The Python programming language's lack of a compiling requirement speeds up the time it takes for students to run and test code.
- The course uses Test Your Code interactive exercises that allow students to practice fixing coding errors.
- Illuminating course graphics provide another way to understand complicated course material.
- Course assignments guide students in improving code they've already written. Other assignments present students with error-ridden code and have them fix the errors in the code. A third type of assignment presents students with a programming challenge and gives instructions to solve that challenge by writing a new program.
- Personal questionnaires with programming professionals provide insight into the ways in which programming skills can be used in different careers.
- A nonprogramming section of the course describes different methods of software development.
- A software development-related assignment asks students to research an open-source software project of their choosing to answer several questions about it.

Grading

Just as with our other e2020 courses, you always have the flexibility to tailor the grade weights for the course according to your own district, school, and student needs.

The default grades for this course are as follows:

- Assignments 20%
- Essays 0%
- Quizzes 20%
- Tests 10%
- Exams 0%
- Projects 50%
- Additional 0%