

INTRODUCTION TO PROGRAMMING

Introduction to Programming is a 70-hour course for students in Grade 8 to provide an understanding of basic computer programming concepts and logic. Students will be introduced to programming through a variety of projects and object-based programming activities and applications. Students will explore and demonstrate business related skills such as teamwork, interpersonal skills and ethics while completing their projects.

Career and technical student organizations (CTSO) are integral, co-curricular components of each career and technical education course. These organizations serve as a means to enhance classroom instruction while helping students develop leadership abilities, expand workplace-readiness skills, and broaden opportunities for personal and professional growth.

Suggested Prerequisite: Computer Essentials

Students will:

Software Development

1. Research differences and similarities between various programming languages.
Examples: C, C++, C#, Java, JavaScript, SQL, PHP (Hypertext Processor), Python, Ruby, etc.
2. Demonstrate comprehension of programming logic.
Examples: creating classes, including methods, arguments, and return values, when do, if then, for next, loops, etc.
3. Construct digital projects using a variety of object-based or language based programming tools.
Examples: Projects - animations, video games, apps, artwork, etc.; Tools - Alice, Kodu Game Lab, Tynker, Scratch, TouchDevelop, etc.
4. Explore opportunities and roles on software development teams.
5. Demonstrate the software development process.
Examples: story board, flow chart, troubleshooting, revision, etc.
6. Explore classifications of computerized entertainment.
7. Explore end user and creative perspective of software development.
8. Demonstrate problem solving and analytical skills while designing projects and correcting programming mistakes.

INTRODUCTION TO PROGRAMMING

Web Design

9. Explain differences among Web site, Web pages, and Web browsers.
Examples: identifying web elements, including hypertext markup language (HTML) tags, headings, and body, etc.
10. Create a business webpage.

Customer Service and Leadership

11. Demonstrate appropriate communication, interpersonal, and teamwork skills when working with customers and co-workers.
Examples: reaction to positive or negative feedback from customer or co-worker, handling complaints, etc.
12. Apply leadership qualities and workplace skills learned through CTSO activities.
13. Explore large-scale software design processes and implementation.
14. Explore global economy and supply chain implementation of software.

Career Opportunities

15. Determine career and entrepreneurial opportunities, responsibilities, educational, and credentialing requirements in the field of computer programming.

Career and Technical Student Organization (CTSO)

16. Utilize programming skills to design digital tools for a local CTSO.
Examples: webpage, mobile app, etc.