Computer Engineering and Technology

Course Credit	1.0
Grade Level(s)	11-12
Prerequisite(s)	Applications of Engineering and Technology

Computer Engineering and Technology is designed to explore the process of taking a software idea and turning it into a profitable product. Students will gain knowledge of the phases of a software life-cycle (planning, design, implementation, testing, deployment, and maintenance). Technology will be introduced by exposing students to industry standard tools for implementing the System Development Life Cycle (SDLC) process. This course will be focused on the SDLC but will expose the students to the various architectures used for a successful project. While not required, it is recommended that students have some prior knowledge of programming languages, databases, operating systems, and platforms.

Foundational standards, shown in the chart below, are an important part of every course. Through these standards, students learn and apply safety concepts, explore career opportunities and requirements, practice the skills needed to succeed in the workplace, learn and practice essential digital literacy skills, develop leadership, and take advantage of the opportunities afforded by Career and Technical Student Organizations (CTSOs). Students in this course may be affiliated with the Technology Student Association (TSA) or Skills USA. The foundational standards are to be incorporated throughout the course.

Foundational Standards	1. Incorporate safety procedures in handling, operating, and maintaining tools and machinery; handling materials; utilizing personal protective equipment; maintaining a safe work area; and following protocols for fire and electrical safety. 2. Demonstrate effective workplace and employability skills, including communication, awareness of diversity, positive work ethic, problem-solving, time management, and teamwork. 3. Explore the range of careers available in the field and investigate their educational requirements, and demonstrate job-seeking skills including resume-writing and interviewing. 4. Demonstrate digital literacy by using digital and electronic tools appropriately, safely, and ethically. 5. Participate in a Career and Technical Student Organization (CTSO) to increase knowledge and skills and to enhance leadership and teamwork.

Computer Engineering and Technology Content Standards

Each content standard completes the stem *"Students will…"*

Planning and Methodology	1. Analyze the various software development methodologies and describe the pros and cons of each one. *Examples: agile, waterfall, spiral, rapid prototype* 2. Collect, document, and decompose all requirements for the completed software system. *Examples: user stories, project requirements document (PRD)* 3. Identify characteristics of a sound financial model to ensure a project can be developed within the projected budget. *Examples: cost, schedule, performance*
Design	4. Analyze various infrastructure options including cloud and in-house hosting for the product. *Examples: cloud, web, mobile, cross platform* 5. Describe software architecture within applications that makes them vulnerable to cyber-attacks. a. Design strategies to counter possible threats to software security. 6. Develop configuration management plans and analyze technologies to manage all work products in designing software. *Examples: software products, documentation, customer feedback*
Implementation	7. Implement scheduling techniques that will ensure adequate time and resources are allocated to deliver a software project on schedule and on budget. 8. Develop metrics and procedures that will ensure all requirements are fully implemented to customer's quality standards. *Examples: peer reviews, QA audits, defect tracking*

Testing	9. Analyze various testing strategies and design test procedures to ensure desired functionality of software products. *Examples: unit testing, regression testing, 508 compliance testing*
Deployment	10. Develop strategies for deploying end products to consumers. *Examples: DevOps, continuous integration*
Maintenance	11. Describe methodologies for tracking defects and planning bug-fix releases. *Examples: Bugzilla, Jira*