Information Technology Fundamentals

Course Credit	1.0
Grade Levels	9-12
Prerequisites	

Information Technology Fundamentals introduces the knowledge base and technical skills for information technology careers. The course presents the basics of computer technology and the functions of information systems. Topics include applications and software, infrastructure, database fundamentals, security, and software development. Emphasis is placed on maintaining a safe working environment and on building technology skills needed for working in the information technology environment.

Career and Technical Student Organizations are integral, co-curricular components of each career and technical education course. These organizations enhance classroom instruction while helping students develop leadership abilities, expand workplace readiness skills, and access opportunities for personal and professional growth. Students in the Information Technology cluster affiliate with SkillsUSA and/or TSA.

Foundational standards, shown in the table 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, develop leadership qualities and take advantage of the opportunities afforded by Career and Technical Student Organizations (CTSOs), and learn and practice essential digital literacy skills. The foundational standards are to be incorporated throughout the course.

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

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 handling hazardous materials and forces.
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. Advocate and practice safe, legal, responsible, and ethical use of information and technology tools specific to the industry pathway.

5. Participate in a Career and Technical Student Organization (CTSO) to increase knowledge and skills and to enhance leadership and teamwork.

6. Use technology to collaborate with peers and/or experts to create digital artifacts that can be published online for a target audience.

7. Formulate new ideas, solve problems, or create products through the design and engineering process by utilizing testing, prototypes, and user feedback.

INFORMATION TECHNOLOGY FUNDAMENTALS
CONTENT STANDARDS

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

Applications and Software

1. Explain the purposes of operating systems, including interfaces between applications and hardware, process management and scheduling, access control protection, and management of applications, memory, disks, and devices.

2. Describe different types of operating systems and explain the advantages and disadvantages of each.
   Examples: mobile vs. computer, proprietary, Linux, Microsoft Operating System

3. Compare and contrast components of operating systems, including file systems, features, file management, services, processes, drivers, utilities, and interfaces.

4. Select and use productivity software for real-world applications.
   Examples: office tools, open source tools

5. Describe various types of applications and delivery models.

   Examples: locally installed, local network hosted, cloud hosted, one-tier, two-tier, three-tier, n-tier, low-code or no-code programming, WYSIWYG web development

6. Configure and manage web browsers, including caching, clearing a cache, deactivating client-side scripting, utilizing browser add-ons and extensions, private browsing, proxy settings, certificates, popup blockers, script blockers, and compatible browsers for various applications.

7. Compare and contrast common data storage units of measurement used for computing.

   Examples: bytes, bits, throughput rate

Infrastructure

8. Compare the purposes of common devices used for networking and peripheral input and output interfaces.

   Examples: scanners, digital cameras, webcams, routers, switches

   a. Distinguish between input and output devices, including monitor, keyboard, mouse, and printer.

      Example: Create a table to categorize devices.

9. Set up and install common peripheral devices to a laptop or desktop PC.

   Examples: external storage, printers, cameras

10. Explain the purposes and functions of common internal computing components.

    Examples: motherboard, hard drive, RAM, expansion card, CPU

11. Compare and contrast the characteristics, advantages, and disadvantages of common Internet service infrastructure, including fiber optic, cable, wireless, and DSL.

12. Compare and contrast cloud computing and traditional computing, including how data elements are organized and where data is stored.

    a. Explain why businesses regard critical data and information as assets.

       Examples: data-driven decisions, crown jewels analysis, trade secrets/proprietary data, patent information

    b. Explain the importance of promoting and protecting the intellectual property of a business.

13. Compare and contrast common computing devices and their purposes.

    Examples: mobile phones, tablets, laptops, servers, game consoles

14. Explain and illustrate basic networking concepts.

    Examples: establishing network communications, inputting device addresses, connecting network devices

15. Summarize and explain the troubleshooting methodology.

16. Install, configure, and secure a basic wireless network.

    Examples: 802.11a/b/g/n/ac standards, modems, routers, cable media

Database Fundamentals

17. Explain the concept of a database and how its use may increase productivity.

    Examples: flowcharts, storage, records, managed database

18. Compare and contrast various database management systems, including structured, semi-structured, and non-structured, and relational and non-relational types.

    Examples: JSon, SQL, XML

19. Design, create, and manage a database structure using various systems.

20. Summarize methods used to interface with databases, including relational, access, and import/export methods.

Security

21. Research and share information on the importance of data confidentiality and security.

22. Explain methods to secure various electronic devices in a network environment.

23. Summarize end-user behavioral security practices.

24. Compare and contrast methods of applying authentication, authorization, accounting, and non-repudiation procedures in a network environment.

25. Explain why an employer may require employees to change passwords regularly.

26. Explain the importance of encryption for data security and describe ways it is commonly used.

27. Explain cybersecurity concepts as they relate to a network.

28. Explain why it is important for businesses to secure and protect their data and describe scenarios which might result in compromised data.
    Examples: human error (social engineering, sharing password), physical compromise of devices (spoofing devices)

Software Development

29. Compare and contrast notational systems.
    Examples: binary, hexadecimal, decimal, ASCII, Unicode

30. Compare and contrast interpreted, compiled, query, and assembly programming language categories.
    Examples: *scripting languages, scripted languages, markup languages*

31. Use programming organizational techniques and demonstrate programming procedures.
    Examples: *scripting languages, scripted languages, markup languages, branching, looping*

32. Explain the purpose and use of programming concepts including identifiers, containers, functions, and objects.

33. Compare and contrast fundamental data types and their characteristics.
    Examples: characters, strings, integers, floats, Boolean

34. Design a step-by-step plan (algorithm) to solve a given problem.
    Example: Recipe for creating brownies from a box mix: Follow the three to five step process written

on the back of the box.

35. Identify decision structures that control program flow.
    Examples: Determine the exact output of a program from a flow chart.
36. Explain techniques for code commenting and documentation.
    Example: inserting meta text in source code

37. Design a program that uses mathematical operations, data, functions, looping and iteration, sequencing, abstraction, lists, and selection.
    Examples: if-else statements, comparison, other operators

Career Opportunities

38. Gather and interpret research data to predict changes in the information technology labor market.