2024 – 2025

COMPUTER SCIENCE EDUCATION COURSES

KENTUCKY CTE
EMPOWERED
Today's Career and Technical Education

COMPUTER SCIENCE EDUCATION COURSES
2024 – 2025

Table of Contents

COMPUTER SCIENCE EDUCATION COURSES 2

- Advanced 3D Game Development 113603 4
- Advanced Game Development and Publishing 113602 5
- AP Computer Science A 110701 7
- AP Computer Science Principles 110711 8
- App Development with Swift 110821 10
- Computational Thinking 110251 11
- Computer Hardware and Software Maintenance 110101 12
- Computer Literacy 110110 13
- Computer Science Co-op 110918 14
- Computer Science Internship 110919 15
- Computer Science Fundamentals 110225 16
- Computers, Networks, and Databases 111001 17
- Cyber Literacy I 110222 18
- Cyber Literacy II 110223 19
- Cyber Science 110224 20
- Cybersecurity 110230 21
- Databases in the Cloud 111003 22
- Design for the Digital World 111002 23
- Design for the Internet 110213 24
- Developing a Cloud Presence 111004 25
- Digital 3D Graphics and Special Effects II 113604 26
- Digital Literacy 060112 27
- Game Design and Development Principles 113605 29
- Help Desk Operations 110102 31
- Internet Technologies 110917 32
- Introduction to Computer Science 110710 33
- Introduction to Database Design 110211 35

Introduction to Digital Game Graphics 113601 36
Introduction to Networking Concepts (non-vendor) 110901 38
Introduction to Programming 110201 39
JAVA Programming I 110205 40
JAVA Programming II 110206 41
JavaScript 110809 42
LAN Switching and Wireless/Scaling Networks/Cisco III 110904 43
Leadership Dynamics – Information Technology 110399 44
Management of Support Services 110302 45
Microsoft Client/Server Configuration 110913 46
Network Fundamentals/Cisco I 110902 47
Network Hardware Installation and Troubleshooting 110906 48
Object-Oriented Programming I 110220 49
Object-Oriented Programming II 110221 50
Productivity Software 110204 51
Project-Based Programming 110226 52
Routing Protocols and Concepts/Cisco II 110903 53
Security Fundamentals 110912 54
Special Topics - Computer Science 110752 55
Special Topics - Information Support and Services 110152 56
Special Topics - Networking 110952 57
Special Topics - Programming 110252 58
Special Topics - Web Development/Administration 110852 59
Web Page Development 110801 60
Web Site Design and Production 110804 61

Advanced 3D Game Development 113603

This course emphasizes creating 3D graphics using one or more state-of-the-art software packages. Instruction provides students with a thorough understanding of techniques for designing advanced 3D games and simulations. Courses will cover 2D and 3D graphics, animation, character development, texturing, rigging, scripting, and game setup using state-of-the-art software development tools. Students spend at least 20 hours of programming and applying learned concepts through programming. (Programming is defined, by the K-12 CS Framework, as the craft of analyzing problems and designing, writing, testing, and maintaining programs to solve them.) Participation in Kentucky Technology Student Association or SkillsUSA will greatly enhance instruction.

Recommended Grade Level: 10 – 12

Recommended Credit: 1

Students will:

1. Compare and contrast modeling methodologies (for example, polygons, NURBS, splines).
2. Explain the applications of low polygon and high polygon construction.
3. Construct and manipulate polygonal objects.
4. Applying texturing/surfacing/shading to models and normal mapping.
5. Identify UVW mapping coordinates.
6. Explain how lighting and shading effect form and surface.
7. Implement basic lighting concepts for ambient and artificial light.
8. Describe the difference between forward and inverse kinematics.
9. Examine the process of particle creation and its application to game design.
10. Create a parent/child hierarchy.
11. Create a joint/bone chain.
12. Apply and adjust weight maps.
13. Create atmospheric effects.
14. Demonstrate the use of constraints to animate objects.
15. Apply various animation techniques (for example, pose-to-pose, straight ahead).
16. Adjust the dynamic properties (for example, gravity, wind speed).
17. Simulate rigid body dynamics (shattering wall, breaking glass).
18. Utilize cinematography in animation.
19. Describe the process of motion capture for animation.

Advanced Game Development and Publishing 113602

This course will focus on creating games using code, 3D characters, objects, and animation utilizing game engines. Students will create work-ready products for the industry. Students will participate in Game Jams to practice working with teams and deadlines. Students spend at least 20 hours of programming and applying learned concepts through programming. (Programming is defined, by the K-12 CS Framework, as the craft of analyzing problems and designing, writing, testing, and maintaining programs to solve them.) Participation in Kentucky Technology Student Association or SkillsUSA will greatly enhance instruction.

Recommended Grade Level: 10 – 12

Recommended Credit: 1

Students will:

1. Compare and contrast licensed vs. proprietary game engines.
2. Debate the strengths and weaknesses of various game engines.
3. Discuss the impact of a game engine on the development of a game.
4. Explain how game engines work.
5. Explain character advancement in relation to storyline and gameplay.
6. Define the size of the player environment.
7. Explain the locations and purpose of non-player characters (NPC).
8. Specify boundaries and borders of the levels within the game.
9. Justify placement of triggers and scripted events.
10. Develop a game with multiple levels.
11. Research types of GUI.
12. Recognize and implement required feedback for the GUI.
13. Create a flowchart that maps the GUI’s functionality.
14. Design and implement a GUI using wireframes.
15. Create a victory condition.
16. Assemble immersive elements into a game.
17. Establish reward systems and in-game economies.
18. Apply game mechanics to the game world.
19. Balance and test game mechanics.
20. Integrate different types of audio including sound effects, ambient background, dialog, and score.
21. Practice creating sound loops.
22. Determine acceptable media files for game development such as sound, graphics, and video.
23. Import appropriate media for a game.
24. Incorporate feedback sounds.
25. Compare and contrast the benefits of various platforms and their target markets.
26. Evaluate the need for flexibility and scalability when developing for a PC.
27. Explore development tools specific to various consoles.
28. Research procedures to deliver a game to mobile markets.
29. Pitch a project and defend why it is entertaining.

30. Explain the role of social media in marketing.
31. Describe crowdsourcing and crowdfunding.
32. Explain the merchandising and branding behind video games.
33. Analyze successful trailers.
34. Explain the concepts of localization and its impact on design.
35. Describe various pay models, such as free-to-play, pay-to-play, singer-user license, and freemium.
36. Describe the integration of social components in a game.
37. Explain the role of social media in the gaming community.
38. Describe professional events in digital gaming.
39. Summarize characteristics of cloud gaming.
40. Evaluate the advances of multi-player gaming.
41. Discuss trends in input devices.
42. Examine current trends in output devices and displays.
43. Explore advances in peripheral devices.

AP Computer Science A 110701

AP Computer Science A introduces students to computer science through programming. Fundamental topics in this course include the design of solutions to problems, the use of data structures to organize large sets of data, the development and implementation of algorithms to process data and discover new information, the analysis of potential solutions, and the ethical and social implications of computing systems. The course emphasizes object-oriented programming and design using the Java programming language. Students spend at least 20 hours of programming and applying learned concepts through programming. (Programming is defined, by the K-12 CS Framework, as the craft of analyzing problems and designing, writing, testing, and maintaining programs to solve them). Participation in Kentucky Technology Student Association or SkillsUSA will greatly enhance instruction.

Recommended Grade Level: 10 – 12

Recommended Credit: 1

Students will:

1. Design, implement and analyze solutions to problems.
2. Use and implement commonly used algorithms.
3. Develop and select appropriate algorithms and data structures to solve new problems.
4. Write solutions fluently in an object-oriented paradigm.
5. Write, run, test, and debug solutions in the Java programming language, utilizing standard Java library classes and interfaces from the AP Java Subset.
6. Read and understand programs consisting of several classes and interacting objects.
7. Read and understand a description of the design and development process leading to such a problem.
8. Understand the ethical and social implications of computer use.

AP Computer Science Principles 110711

AP Computer Science Principles introduces students to the breadth of the field of computer science. In this course, students will learn to design and evaluate solutions and to apply computer science to solve problems through the development of algorithms and programs. They will incorporate abstraction into programs and use data to discover new knowledge. Students will also explain how computing innovations and computing systems, including the Internet, work, explore their potential impacts and contribute to a computing culture that is collaborative and ethical. Students spend at least 20 hours of programming and applying learned concepts through programming. (Programming is defined, by the K-12 CS Framework, as the craft of analyzing problems and designing, writing, testing, and maintaining programs to solve them). Participation in Kentucky Technology Student Association or SkillsUSA will greatly enhance instruction.

Recommended Grade Level: 9 – 12

Recommended Credit: 1

Students will:

1. Use computing tools and techniques to create artifacts.
2. Collaborate in the creation of computational artifacts.
3. Analyze computational artifacts; use computing tools and techniques for creative expression.
4. Use programming as a creative tool.
5. Describe the combination of abstractions used to represent data.
6. Explain how binary sequences are used to represent digital data.
7. Develop and abstraction.
8. Use multiple levels of abstraction in computation.
9. Use models and simulations to raise and answer questions.
10. Use computers to process information to gain insight and knowledge.
11. Collaborate when processing information to gain insight and knowledge.
12. Communicate insight and knowledge gained from using computer programs to process information.
13. Use computing to facilitate exploration and the discovery of connections in information.
14. Use large data sets to explore and discover information and knowledge.
15. Analyze the considerations involved in the computational manipulation of information.
16. Develop an algorithm designed to be implemented to run on a computer.
17. Express an algorithm in a language.
18. Appropriately connect problems and potential algorithmic solutions.
19. Evaluate algorithms analytically and empirically.
20. Explain how programs implement algorithms.
21. Use abstraction to manage complexity in programs.
22. Evaluate a program for correctness.
23. Develop a correct program.
24. Collaborate to solve a problem using programming.
25. Employ appropriate mathematical and logical concepts in programming.

26. Explain the abstractions on the Internet and how the Internet functions.
27. Explain characteristics of the Internet and systems built on it.
28. Analyze how characteristics of the Internet and the system built on it influence their use.
29. Connect the concern of cybersecurity with the Internet and the systems built on it.
30. Analyze how computing affects communication, interaction, and cognition.
31. Collaborate as part of a process that scales.
32. Connect computing with innovations in other fields.
33. Analyze the beneficial and harmful effects of computing.
34. Connect computing within the economic, social, and cultural context.

App Development with Swift 110821

App Development with Swift allows students to build a foundation in Swift, UIKit, and networking through hands-on labs and guided projects. At the end of each of the first five units, students will complete guided projects. Through these projects, students will create features that interest them, all while performing the type of work they can expect in an app development workplace. In the last unit, students will examine how to design, prototype, and architect an app of their own design. Students spend at least 20 hours of programming and applying learned concepts through programming. (Programming is defined, by the K-12 CS Framework, as the craft of analyzing problems and designing, writing, testing, and maintaining programs to solve them.) Participation in Kentucky Technology Student Association or SkillsUSA will greatly enhance instruction.

Recommended Grade Level: 9 – 12

Recommended Credit: 1

Students will:

1. Explain the basics of data, operators, and control flow in Swift.
2. Explain the basics of documentation, debugging, Xcode, building and running an app, and Interface Builder.
3. Apply this knowledge to create a simple flashlight app.
4. Explore Swift strings, functions, structures, collections, and loops.
5. Explore UIKit and how to display data using Auto Layout and stack views.
6. Practice this knowledge to build a word-guessing game app.
7. Discover how to build simple workflows and navigation hierarchies using navigation controllers, tab bar controllers, and segues.
8. Examine two powerful tools in Swift, optionals and enumerations.
9. Apply this knowledge into practice with a personalized survey that reveals a response to the user.
10. Discover scroll views, table views, and building complex input screens.
11. Explore how to save data, share data with other apps, and work with images in the user’s photo library.
12. Practice these skills with a task-tracking app that allows the user to add, edit, and delete items in a familiar table-based interface.
13. Customize the task-tracking app to keep track of any type of information.
14. Determine how to use animations, concurrency, and how to work with the web.
15. Apply this knowledge with a customizable menu app that displays the available dishes from a restaurant and allows the user to submit an order.
16. Use a web service that allows students to set up a menu with customized menu items and photos.
17. Design, prototype, and architect a project of their own design.
18. Build this project independently.

Computational Thinking 110251

Computational Thinking promotes understanding of computer programming and logic by teaching students to think like a computer. It covers skills needed to develop and design language-independent solutions to solve computer-related problems. Instruction covers the development and design basics including the use of variables, control and data structures, and principles of command-line and object-oriented languages. Students spend at least 20 hours of programming and applying learned concepts through programming. (Programming is defined, by the K-12 CS Framework, as the craft of analyzing problems and designing, writing, testing, and maintaining programs to solve them.) Participation in Kentucky Technology Student Association or SkillsUSA will greatly enhance instruction.

Recommended Grade Level: 10 – 12

Recommended Credit: 1

Students will:

1. Demonstrate an understanding of elementary logic, truth tables, and Boolean algebra.
2. Demonstrate programming style best practices.
3. Illustrate the flow of a program.
4. Illustrate concepts using one or more programming languages.
5. Explain the implications of file processing.
6. Describe the steps addressed in the design of a program to solve the state problem.
7. Explain how algorithms are used to produce artificial intelligence (AI)
8. Describe the principles of object-oriented programming.
9. Develop algorithms with an increasing degree of complexity using structured programming techniques such as sequence, selection, and repetition.
10. Use fundamental data types and data structures such as integers, reals, characters, strings, Booleans, one - and two - dimensional arrays.
11. Analyze the binary representation of data.
12. Use modular programming.

Computer Hardware and Software Maintenance 110101

This course presents a practical view of computer hardware and client operating systems. It also covers computer hardware components; troubleshooting, repair, and maintenance; operating system interfaces and management tools; networking components; computer security; and operating procedures. Students spend at least 20 hours of programming and applying learned concepts through programming. (Programming is defined, by the K-12 CS Framework, as the craft of analyzing problems and designing, writing, testing, and maintaining programs to solve them.) Participation in Kentucky Technology Student Association or SkillsUSA will greatly enhance instruction.

Recommended Grade Level: 9 – 12

Recommended Credit: .5 – 1

Students will:

1. Identify and explain motherboard components.
2. Identify, install, configure, and upgrade personal computer components.
3. Perform device driver installation, scheduled maintenance, and memory and firmware updates.
4. Identify common tools, basic diagnostic procedures, troubleshooting techniques, and preventative maintenance methods.
5. Explain and apply the troubleshooting process to diagnose and repair common hardware and software problems.
6. Demonstrate an understanding of conversion between binary, decimal, hexadecimal number systems.
7. Compare and contrast client operating systems and their features.
8. Use multiple user interfaces, including command-line, to perform operating system management tasks, to configure, optimize and upgrade the current client operating system and to diagnose network connection issues.
9. Use and manage file systems, operating system utilities, backup programs, and optimization tools.
10. Describe the process to install, configure, secure and troubleshoot a basic small or home office network.
11. Identify the fundamental principles of networking and security.
12. Describe and apply appropriate operational procedures including safety, environmental procedures, good communication skills, and professional behavior.

Computer Literacy 110110

This course provides an introduction to the computer and the convergence of technology as used in today's global environment. Introduces topics including computer hardware and software, file management, the Internet, e-mail, social web, green computing, AR and VR, security, and AI fundamentals. Instruction presents the basic use of application, programming, systems, and utility software. Students spend at least 20 hours of programming and applying learned concepts through programming. (Programming is defined, by the K-12 CS Framework, as the craft of analyzing problems and designing, writing, testing, and maintaining programs to solve them.) Participation in Kentucky Technology Student Association or SkillsUSA will greatly enhance instruction.

Recommended Grade Level: 9 – 12

Recommended Credit: 1

Students will:

1. Describe basic computer functions and use correct computer terminology.
2. Use a course management system.
3. Utilize computer technology as a tool to access, manage, prepare, and present information.
4. Identify trends in information processing and new emerging technologies.
5. Explain the impact of computers upon society including effects of social technologies, green computing, dangers of excessive use, and disposal of obsolete equipment.
6. Identify and analyze ethical issues such as copyright, privacy, and security as related to computing.
7. Explain the difference between application, programming, system, and utility software.
8. Use a graphical user interface-based operating system to manage files, folders and disks.
9. Use application software packages to prepare basic documents, spreadsheets, databases, and presentations.
10. Describe and explain basic data communications and network technologies and functions.
11. Identify and use basic e-mail and Internet functions and understand their capabilities.
12. Describe globalization and challenges including technological barriers, electronic payments, and varying cultures.
13. Describe cloud computing and its impact on business and personal systems.
14. Explain the importance of maintaining a good digital identity.
15. Explain how Cellular service differs from Internet service.
16. Explore the impact of AI on society.

Computer Science Co-op[*] 110918

Cooperative Education for CTE courses provides supervised worksite experience related to the student’s identified career pathway. A student must be enrolled in an approved course during the same school year that the co-op experience is completed. Students who participate receive a salary for these experiences, in accordance with local, state and federal minimum wage requirements according to the Work-Based Learning Manual. Students spend at least 20 hours of programming and applying learned concepts through programming. (Programming is defined, by the K-12 CS Framework, as the craft of analyzing problems and designing, writing, testing, and maintaining programs to solve them.)

Recommended Grade Level: 11 – 12

Recommended Credit: 1

Students will:

1. Demonstrate and practice safe work habits at all times.
2. Gain career awareness and the opportunity to test career choices.
3. Receive work experience related to career interests.
4. Integrate classroom studies with work experiences.
5. Receive exposure to the facilities and equipment unavailable in a classroom setting.
6. Increase employability potential.

[*] Co-op can only be taken after the first four credits are earned, OR along with another course in the pathway, OR if the student is enrolled in an approved pre-apprenticeship program

Computer Science Internship 110919

Internship for CTE courses provides supervised worksite experience for high school students who are enrolled in a course associated with their identified career pathway. Internship experiences consist of a combination of classroom instruction and field experiences. A student receiving pay for an intern experience is one who is participating in an experience that lasts a semester or longer and has an established employee- employer relationship. A non-paid internship affects those students who participate on a short-term basis (semester or less). All information references to the Work-Based Learning Manual. Students spend at least 20 hours of programming and applying learned concepts through programming. (Programming is defined, by the K-12 CS Framework, as the craft of analyzing problems and designing, writing, testing, and maintaining programs to solve them.)

Recommended Grade Level: 11 – 12

Recommended Credit: 1

Students will:

1. Demonstrate and practice safe work habits at all times.
2. Gain career awareness and the opportunity to test career choices.
3. Receive work experience related to career interests.
4. Integrate classroom studies with work experience.
5. Receive exposure to facilities and equipment unavailable in a classroom setting.
6. Increase employability potential.

Computer Science Fundamentals 110225

Computer Science is a project-driven, application-based course. Computer Science engages students in an immersive exploration of the breadth of the field of computer science. Students design and create programs, construct simple circuits, and discuss the impacts of computer science in society. Students spend at least 20 hours of programming and applying learned concepts through programming. (Programming is defined, by the K-12 CS Framework, as the craft of analyzing problems and designing, writing, testing, and maintaining programs to solve them.) Participation in Kentucky Technology Student Association or SkillsUSA will greatly enhance instruction.

Recommended Grade Level: 9 – 12

Recommended Credit: 1

Students will:

1. Develop algorithms as solutions to problems while discovering the importance of understanding problems in order to develop efficient step-by-step solutions.
2. Explore the reasons for using computers to execute problem solutions.
3. Translate algorithms to a language computers can understand.
4. Explore various major data manipulation and processing structures used by computers.
5. Explore and utilize structures to design algorithms to solve problems.
6. Apply computer architecture including using a Raspberry Pi platform to learn how computer hardware provides a powerful platform on which to run the software.

Computers, Networks, and Databases 111001

This project-based learning course engages students who are curious about data science. In this course, students will learn how to use a design process to create systems that acquire, store and communicate data for a variety of career fields. Students will work collaboratively in teams to design systems, solve problems, think critically, be creative, and communicate with each other and business partners. Students spend at least 20 hours of programming and applying learned concepts through programming. (Programming is defined, by the K-12 CS Framework, as the craft of analyzing problems and designing, writing, testing, and maintaining programs to solve them.) Participation in Kentucky Technology Student Association or SkillsUSA will greatly enhance instruction.

Recommended Grade Level: 9 – 12

Recommended Credit: 1

Students will:

1. Use the technical design process to design, build, and test prototypes.
2. Use the terminology of the field.
3. Use data and informatics tools to make decisions and solve problems.
4. Apply project management principles.
5. Use appropriate and effective research skills.
6. Demonstrate proficiency in word processing, spreadsheets, databases, and presentation software.
7. Communicate information, including descriptive statistics, to various stakeholder groups.

Cyber Literacy I 110222

Cyber Literacy I is a hands-on course that builds a strong cyber foundation for high school students. The course introduces students to cyber by blending robotics, programming, electricity, and elements of liberal arts. Students learn about the opportunities, threats, responsibilities, and legal constraints associated with operative in cyberspace. Throughout the course, students learn the basics of electricity, programming, and networking as well as develop critical thinking skills. Cyber Literacy I lays a foundation for further exploration into STEM and cyber-related topics. Students spend at least 20 hours of programming and applying learned concepts through programming. (Programming is defined, by the K-12 CS Framework, as the craft of analyzing problems and designing, writing, testing, and maintaining programs to solve them.) Participation in Kentucky Technology Student Association or SkillsUSA will greatly enhance instruction.

Recommended Grade Level: 9 – 12

Recommended Credit: 1

Students will:

1. Apply the fundamentals of electricity including the basic movement of electrons and experiments that include chemistry, circuitry, and magnetism.
2. Apply BASIC Programming including basic coding essentials through flowcharts, the use of simple programming languages, and the use of simple programming tasks.
3. Engage with a microcontroller as a platform for learning robotics fundamentals.
4. Assemble robots to perform various functions through the implementation of sensors.
5. Apply programming knowledge including the use of autonomous devices, the use of programming components such as conditional and unconditional loops, subroutines, and variable manipulation.
6. Relate electrical components like LEDs, piezo-crystal elements, infrared light, and tactile sensors to cyber.
7. Illustrate real-world applications and implications of computers and the internet in our society today.
8. Deliberate the historical and societal context of cyber.
9. Engage in a lab and project-driven lesson regarding learning about cyberspace.
10. Engage in a lab and project-driven lesson regarding the ethical concerns about online behavior, cyberbullying, and cybersecurity.
11. Engage in a lab and project-driven lesson regarding the ethical concerns about designing autonomous devices and artificial intelligence.

Cyber Literacy II 110223

Cyber Literacy II is a project-driven course that expands a student's understanding of cyberspace through two primary topics: systems engineering and liberal arts. The Cyber Literacy II course builds upon fundamental cyber skills. Many aspects of science, engineering, technology, and mathematics are included throughout the course. Students are challenged to create flowcharts with each build as well as read schematics instead of relying on wiring diagrams. Students spend at least 20 hours of programming and applying learned concepts through programming. (Programming is defined, by the K-12 CS Framework, as the craft of analyzing problems and designing, writing, testing, and maintaining programs to solve them.) Participation in Kentucky Technology Student Association or SkillsUSA will greatly enhance instruction.

Recommended Grade Level: 9 – 12

Recommended Credit: 1

Students will:

1. Apply systems engineering to build a variety of multi-component projects.
2. Employ flowcharts to discuss data flow and pseudo-code.
3. Engage in a variety of labs and project-driven lessons that prompt learning regarding robotics.
4. Combine components onto a platform to design autonomous systems while considering the real-world impact of the systems and how they impact the students' environment.
5. Synthesize content with issues including privacy, security, and technology.
6. Synthesize content with issues including search warrants, digital media, and the requirements to obtain a search warrant.
7. Synthesize content with issues including cyberbullying and real-world examples of the implications of cyberbullying.
8. Defend a position in debates on national security.
9. Engage in a variety of labs and project-driven lessons that prompt learning regarding the impact and relationship between expectations of privacy and security.

Cyber Science 110224

Cyber Science is an innovative, project-driven course that integrates science, technology, engineering, and mathematics (STEM) disciplines with liberal arts. Throughout the course, students are engaged in a systems-level approach to problem-solving using robotics and computer science in the context of liberal arts. Seamlessly integrating the different disciplines provides students with a dynamic learning environment and a unique educational experience. Through Cyber Science, students are not only able to make meaningful connections between STEM and liberal arts, but they also learn how to become better cyber citizens. This course introduces Networking and Security, Programming Basics, Ethics and Societal Issues, Foundations of Computer Science,and Artificial Intelligence. Students spend at least 20 hours of programming and applying learned concepts through programming. (Programming is defined, by the K-12 CS Framework, as the craft of analyzing problems and designing, writing, testing, and maintaining programs to solve them.) Participation in Kentucky Technology Student Association or SkillsUSA will greatly enhance instruction.

Recommended Grade Level: 9 – 12

Recommended Credit: 1

Students will:

1. Apply programming basics including developing programming skills through a progression of robot activities.
2. Apply foundations of computer science including Boolean logic, variables, flow charts, data structures, and sorting using robot applications.
3. Apply networking and security foundations including showcasing the structure of networks as well as the vulnerabilities.
4. Demonstrate the need for security through emphasizing man-in-the-middle attacks, cryptography, and stenography.
5. Utilize artificial intelligence including applying the concepts of heuristics.
6. Utilize sensors to read input in order to produce the desired output through various robot projects.
7. Explore the historical, ethical, and societal impacts of cyber.
8. Defend a position in debates regarding cyber.

Cybersecurity 110230

Cybersecurity introduces the tools and concepts of cybersecurity and encourages students to create solutions that allow people to share computing resources while protecting privacy. This course raises students' knowledge of and commitment to ethical computing behavior. Students will learn the components of cybersecurity and the role each plays in preventing, detecting, and mitigating vulnerabilities and attacks. Students spend at least 20 hours of programming and applying learned concepts through programming. (Programming is defined, by the K-12 CS Framework, as the craft of analyzing problems and designing, writing, testing, and maintaining programs to solve them.) Participation in Kentucky Technology Student Association or SkillsUSA will greatly enhance instruction.

Recommended Grade Level: 9 – 12

Recommended Credit: 1

Students will:

1. Identify the goals, objectives and purposes of cybersecurity
2. Research the history of the Internet and its impact on government, society and business.
3. Describe the concepts of malware attack vectors
4. Mitigate threats by remaining abreast of industry information
5. Explain basic security concepts
6. Identify issues that affect physical and remote access device security
7. Explain data and privacy encryption issues related to using technology
8. Compare and contrast physical security controls
9. Explain the purpose of various network access control models
10. Identify common network vulnerabilities, threats, and risks
11. Explain the functions and application of various network devices
12. Describe the need for security safeguards
13. Demonstrate the components of cybersecurity and the role each plays in preventing and detecting attacks
14. Assess the role of strategy and policy in determining the success of information security.
15. Contrast the various approaches to security training and formulate a simple training agenda
16. Evaluate the trends and patterns that will determine the future state of cybersecurity.

Databases in the Cloud 111003

This project-based learning course is for students who want to tackle the more complex challenges that business and industry face. Students at this level will learn about Web technologies, cloud storage, information security, data, animations, introductory computer programming, and database applications. Students will take more responsibility for their own learning, problem-solving, and thinking outside of the box. Real-world challenges will require higher levels of research, building, testing, analyzing, and improving systems. Students spend at least 20 hours of programming and applying learned concepts through programming. (Programming is defined, by the K-12 CS Framework, as the craft of analyzing problems and designing, writing, testing, and maintaining programs to solve them.) Participation in Kentucky Technology Student Association or SkillsUSA will greatly enhance instruction.

Recommended Grade Level: 9 – 12

Recommended Credit: 1

Students will:

1. Use the terminology of the field.
2. Research data science technical texts, journal articles, and other related documents in developing a plan.
3. Use the five-step software/system life cycle (design, build, test, implement, and evolve).
4. Use data science concepts to solve problems.
5. Use data and data science tools to make decisions and solve problems.
6. Apply project management principles.
7. Gain information on how the American computer industry works.
8. Use appropriate and effective research skills.
9. Use best practices to design and implement research studies.
10. Use the scientific method to design investigations.
11. Demonstrate proficiency in word processing, spreadsheets/databases, and presentation software.
12. Communicate information, including descriptive statistics, to various audiences.

Design for the Digital World 111002

This project-based learning course engages students who are interested in applying the design process to create systems such as a cloud-based digital storage system for images. Students will design a system to automatically collect and report data on highway usage. They will apply a geospatial system to map a store and develop a database that studies shopping habits. Through these projects, students will learn about data management and logic-based queries by collecting data, using the Global Positioning Systems (GPS), and analyzing data utilizing a geographic information system (GIS). They will learn how to automate data collection to make processes more effective and efficient. Students will work collaboratively in teams and demonstrate their knowledge and skills by presenting new and innovative ideas, techniques, and solutions to business and industry partners. Students spend at least 20 hours of programming and applying learned concepts through programming. (Programming is defined, by the K-12 CS Framework, as the craft of analyzing problems and designing, writing, testing, and maintaining programs to solve them.) Participation in Kentucky Technology Student Association or SkillsUSA will greatly enhance instruction.

Recommended Grade Level: 9 – 12

Recommended Credit: 1

Students will:

1. Use the terminology of the field.
2. Research data science technical texts, journal articles, and other related documents in developing a plan.
3. Use the five-step software/system life cycle (design, build, test, implement, and evolve).
4. Use data science concepts to solve problems.
5. Use data and data science tools to make decisions and solve problems.
6. Apply project management principles.
7. Gain information on how the American computer industry works.
8. Use appropriate and effective research skills.
9. Use best practices to design and implement research studies.
10. Use the scientific method to design investigations.
11. Demonstrate proficiency in word processing, spreadsheets/databases, and presentation software.
12. Communicate information, including descriptive statistics, to various audiences.

Design for the Internet 110213

This course introduces basic computer graphics with special emphasis on graphics for games. Students spend at least 20 hours of programming and applying learned concepts through programming. (Programming is defined, by the K-12 CS Framework, as the craft of analyzing problems and designing, writing, testing, and maintaining programs to solve them.) Participation in Kentucky Technology Student Association or SkillsUSA will greatly enhance instruction.

Recommended Grade Level: 10 – 12

Recommended Credit: 1

Students will:

1. Identify the principles of communication through the visual medium using text, still imagery and video technology.
2. Explain copyright laws affecting digital graphics including images and image use.
3. Identify the purpose of, audience, storyboarding, and audience needs for preparing images.
4. Explain the design process for various forms of digital media.
5. Identify considerations of designing for a specific audience, including paid customers.
6. Analyze and evaluate digital media content for audience, purpose and design techniques.
7. Identify trends in the use and creation of digitally generated media.
8. Explain the key elements of drawing and painting.
9. Explain image resolution, image size, and image file format for the web, video, and print.
10. Demonstrate effective message composition and design using industry-standard design elements and principles.
11. Explain the principles of image composition.
12. Explain digital typography.
13. Differentiate between typeface and font.
14. Demonstrate digital camera and scanner operation.
15. Define digital image terminology.
16. Explain image and editing layers.
17. Demonstrate importing, exporting, organizing, and saving digital graphic files.
18. Manipulate image selections and measurement.
19. Use digital graphic editing software guides and rulers.
20. Transform digital images using editing applications.
21. Adjust or correct the tonal range, color, or distortions of an image using editing applications.
22. Explain retouching and blending images.
23. Explain and apply digital image editing filters.
24. Prepare images for web, print, and video.
25. Identify career and entrepreneurial opportunities in digital graphics technology.

Developing a Cloud Presence 111004

Students in this course will focus on the ethics of privacy, social networking, designing for clients and artificial intelligence through authentic projects. Students will select a business partner and design, build and test a Web presence for a company . Student teams will work collaboratively with a business partner to develop a proposal for the project with evaluation criteria. Once the business partner accepts the proposal, the student team will implement it by designing, planning, building the system, and testing and revising the system to meet the needs of the business. Students spend at least 20 hours of programming and applying learned concepts through programming. (Programming is defined, by the K-12 CS Framework, as the craft of analyzing problems and designing, writing, testing, and maintaining programs to solve them.) Participation in Kentucky Technology Student Association or SkillsUSA will greatly enhance instruction.

Recommended Grade Level: 9 – 12

Recommended Credit: 1

Students will:

1. Use the terminology of the field.
2. Research data science technical texts, journal articles, and other related documents in developing a plan.
3. Use the five-step software/system life cycle (design, build, test, implement, and evolve).
4. Use data science concepts to solve problems.
5. Use data and data science tools to make decisions and solve problems.
6. Apply project management principles.
7. Gain information on how the American computer industry works.
8. Use appropriate and effective research skills.
9. Use best practices to design and implement research studies.
10. Use the scientific method to design investigations.
11. Demonstrate proficiency in word processing, spreadsheets, databases, and presentation software.
12. Communicate information, including descriptive statistics, to various audiences.

Digital 3D Graphics and Special Effects II 113604

This course will focus on creating games using code, 3D characters, objects, and animation utilizing game engines. Students will create work-ready products for the industry. Introduces advanced texturing and lighting techniques to enhance depth perception and realism within 3D environments. Students spend at least 20 hours of programming and applying learned concepts through programming. (Programming is defined, by the K-12 CS Framework, as the craft of analyzing problems and designing, writing, testing, and maintaining programs to solve them.) Participation in Kentucky Technology Student Association or SkillsUSA will greatly enhance instruction.

Recommended Grade Level: 9 – 12

Recommended Credit: 1

Students will:

1. Demonstrate an understanding of how to add textures to objects.
2. Use appropriate types of lighting techniques to design.
3. Demonstrate adding depth using different types of shadowing techniques.
4. Create custom connections and color utilities to innovative designs.
5. Use indirect and direct illumination to designs.

Digital Literacy 060112

Students will use a computer and application software including word processing, presentation, database, spreadsheet, internet, and email to prepare elementary documents and reports. The impact of computers on society and ethical issues are presented. Leadership development will be provided through FBLA and/or DECA.

Recommended Grade Level: 9 – 12

Recommended Credit: 1

Students will:

1. Use a word processing program to create, save, print, modify, spell-check, and grammar-check a simple document.
2. Use a word processing program to enhance the appearance of a simple document by using centered, right justification, boldface, underlined, and italicized text.
3. Use a word processing program to change the default margins and line spacing.
4. Use a word processing program to create a document with headers, footers, and footnotes.
5. Use a presentation program with text body, graphics, and animation.
6. Use an electronic spreadsheet program to create, save, print, modify, and obtain graphs from a simple spreadsheet.
7. Use an electronic spreadsheet program to perform basic mathematical operations including, but not limited to, addition, subtraction, multiplication and division.
8. Use an electronic spreadsheet program to calculate averages and percentages.
9. Use an electronic spreadsheet program to enhance the appearance of a spreadsheet by changing fonts, foreground, and background colors and centering text across columns.
10. Use a database management program to create, maintain, and print reports from a simple relational database.
11. Use a database management program to customize the user interface by creating and maintaining forms and reports.
12. Use a database management program to query tables using basic query operations such as “and”, “or”, “not”.
13. Print in landscape and portrait orientations.
14. Use the component of the operating system that helps the user manipulate files and folders to copy, move, rename, and delete files and to create, copy, move, rename, and delete folders.
15. Use the World Wide Web browser to navigate hypertext documents and to download files.
16. Use Internet search engines and understand their advantages and disadvantages.
17. Use an electronic mail program to send and receive electronic mail.
18. Identify components of a computer.
19. Discriminate between ethical and unethical uses of computers and information.
20. Demonstrate a basic understanding of issues regarding software copyright, software licensing, and software copying.

21. Demonstrate an awareness of computer viruses and a basic understanding of ways to protect a computer from viruses.
22. Demonstrate a basic understanding of the impact of computers on society.
23. Use and understand basic computer terminology.

Game Design and Development Principles 113605

This course is an introduction to Game Design and Gaming. The course provides an overview of story development, gaming history, game reviews, current gaming trends, and industry software. Students will begin to create and develop a game story/plot that can be further developed in higher-level courses as well as critique current games. In addition, 2D game development software and image manipulation will be explored to further enhance their design skills. Career exploration into game design will be researched and gain awareness of job and postsecondary opportunities. Students spend at least 20 hours of programming and applying learned concepts through programming. (Programming is defined, by the K-12 CS Framework, as the craft of analyzing problems and designing, writing, testing, and maintaining programs to solve them.) Participation in Kentucky Technology Student Association or SkillsUSA will greatly enhance instruction.

Recommended Grade Level: 9 – 12

Recommended Credit: 1

Students will:

1. Explain the history of computing technologies that impact the game development industry.
2. Explore non-digital games.
3. Research the evolution of video games.
4. Describe the different game genres.
5. Evaluate contributions of individual game designers and developers.
6. Explore careers as a game artist and sound designer.
7. Describe the role of game designer.
8. Explore careers as a game developer.
9. Describe career pathways in quality assurance/testing.
10. Explain the role of the producer.
11. Explain the career path of an independent developer.
12. Research salary structures in the industry.
13. Define common terminology and its acronyms.
14. Identify the tools to develop a game (engine, application program interface [API], digital content creation tools, editors).
15. Communicate both in writing and verbally using appropriate industry terminology.
16. Compare and contrast the entertainment software rating boards (ESRB) ratings for games.
17. Explain the principles of visual design.
18. Explain the elements of design.
19. Analyze artwork/designs for specific design theories.
20. Explore the components of game structure.
21. Analyze the essentials of storytelling.
22. Write an outline of a nonlinear story.
23. Create rules for a game.
24. Compare conflict and outcomes.

25. Develop objectives and outcomes for a game.
26. Explain the importance of usability and how it impacts user experience.
27. Explain in-game economies, motivators, and reward systems.
28. Research various styles of game documentation.
29. Develop a technical design document (TDD).
30. Describe components of a game design document (GDD).
31. Produce a game design document.
32. Produce a game pitch document.
33. Present game documentation.
34. Compare and contrast categories of game mechanics.
35. Research victory condition mechanics of a game.
36. Discuss the relationship between game mechanics and game complexity and interactions.
37. Incorporate game mechanics into a game.
38. Explain basic logic statements such as if/then and cause/effect.
39. Describe uses of Boolean operators and symbols associated with them.
40. Generate truth tables for game events.
41. Examine different number systems including binary, decimal, and hexadecimal.
42. Demonstrate proper use of the order of operations.
43. Convert mathematical formulas into code.
44. Explain when to apply mathematical concepts common to game coding.
45. Use logical thinking to create a diagram of code execution.
46. Research laws that govern intellectual property in diverse forms.
47. Evaluate Creative Commons and open source licensure.
48. Cite the boundaries of third-party work.
49. Explain copyright, trademarks, and other intellectual property protection.
50. Explain invasion of privacy in the use of technology.
51. Model acceptable security practices.
52. Explore the issues of piracy and digital rights management (DRM).
53. Analyze your personal digital footprint.
54. Discuss social responsibility and issues concerning gaming.
55. Model legal and ethical use of information.
56. Identify key elements of non-disclosure agreements (NDA) and contracts.
57. Summarize the behavior of an algorithm.
58. Determine if a given algorithm successfully solves a stated problem.

Help Desk Operations 110102

Help Desk Operations introduces a variety of tools and techniques to provide user support in help desk operations. The course explores help desk concepts, customer service skills, troubleshooting problems, writing for end users, help desk operations and software, needs analysis, facilities management, and other topics related to end-user support. Students spend at least 20 hours of programming and applying learned concepts through programming. (Programming is defined, by the K-12 CS Framework, as the craft of analyzing problems and designing, writing, testing, and maintaining programs to solve them.) Participation in Kentucky Technology Student Association or SkillsUSA will greatly enhance instruction.

Recommended Grade Level: 9 – 12

Recommended Credit: 1

Students will:

1. Define the role of help desk and customer service in an organization.
2. Evaluate help desk technology, tools, and techniques.
3. Identify common support problems, including software tools and features.
4. Identify service technology trends.
5. Demonstrate professional and effective communication skills.
6. Demonstrate team-building strategies.
7. Develop technical training materials, and other user documentation to support help desk operations.
8. Demonstrate a methodical approach to the problem-solving process.
9. Apply conflict resolution techniques and skills in customer support.
10. Exhibit positive professionalism with customers and technical writing skills.
11. Demonstrate personal, system, and stress management by way of using self-help tools.
12. Use support performance and reporting tools, call management software, problem resolution software, asset and change management tools, and notification tools for support in additional level two and level three support tools.

Internet Technologies 110917

This course provides students with a study of traditional and emerging Internet technologies. Also covered are other topics including Internet fundamentals, Internet applications, Internet delivery systems, and Internet client/server computing. Internet Technologies provides a hands-on experience and some rudimentary programming in an Internet environment. Students spend at least 20 hours of programming and applying learned concepts through programming. (Programming is defined, by the K-12 CS Framework, as the craft of analyzing problems and designing, writing, testing, and maintaining programs to solve them.) Participation in Kentucky Technology Student Association or SkillsUSA will greatly enhance instruction.

Recommended Grade Level: 10 – 12

Recommended Credit: 1

Students will:

1. Describe the history of the Internet and its impact on government, society, and business.
2. Describe the models used to organize Internet technologies.
3. Explain how the Internet is governed and the standards that are used.
4. Describe the protocols that make the Internet work.
5. Use Internet technologies for data transfer, remote access, information delivery, email, content presentation, and real-time collaboration.
6. Describe how the Internet is used for e-commerce.
7. Describe Internet naming conventions, URLs, and web server file organization.
8. Describe core connectivity issues such as NAT, ISP’s, and IP addresses.
9. Create and publish simple web content using basic HTML.
10. Use existing scripting applications and create simple client/server applications to enhance information delivery.

Introduction to Computer Science 110710

Introduction to Computer Science is designed to introduce students to the breadth of the field of computer science through an exploration of engaging and accessible topics. Rather than focusing the entire course on learning particular software tools or programming languages, the course is designed to focus on the conceptual ideas of computing and help students understand why certain tools or languages might be utilized to solve particular problems. The goal of the course is to develop in students the computational practices of algorithm development, problem-solving, and programming within the context of problems that are relevant to the lives of today's students. Students will also be introduced to topics such as interface design, limits of computers, and societal and ethical issues. Students spend at least 20 hours of programming and applying learned concepts through programming. (Programming is defined, by the K-12 CS Framework, as the craft of analyzing problems and designing, writing, testing, and maintaining programs to solve them.) Participation in Kentucky Technology Student Association or SkillsUSA will greatly enhance instruction.

Recommended Grade Level: 9 – 12

Recommended Credit: 1

Students will:

1. Analyze the characteristics of hardware components to determine the applications for which they can be used.
2. Use appropriate tools and methods to execute Internet searches that yield requested data.
3. Evaluate the results of web searches and the reliability of information found on the Internet.
4. Explain the differences between tasks that can and cannot be accomplished with a computer.
5. Analyze the effects of computing on society within economic, social, and cultural contexts.
6. Communicate legal and ethical concerns raised by computing innovation.
7. Explain the implications of communication as data exchange.
8. Name and explain the steps they use in solving a problem.
9. Solve a problem by applying appropriate problem-solving techniques.
10. Express a solution using standard design tools.
11. Determine if a given algorithm successfully solves a stated problem.
12. Create algorithms that meet specified objectives.
13. Explain the connections between binary numbers and computers.
14. Summarize the behavior of an algorithm.
15. Compare the tradeoffs between different algorithms for solving the same problem.
16. Explain the characteristics of problems that cannot be solved by an algorithm.
17. Use appropriate algorithms to solve problems.
18. Design, code, test, and execute a program that corresponds to a set of specifications.

19. Select appropriate programming structures.
20. Locate and correct errors in a program.
21. Explain how a particular program functions.
22. Justify the correctness of a program.
23. Create programs with practical, personal, and/or societal intent.
24. Describe the features of appropriate data sets for specific problems.
25. Apply a variety of analysis techniques to large data sets.
26. Use computers to find patterns in data and test hypotheses about data.
27. Compare different analysis techniques and discuss the tradeoffs among them.
28. Justify conclusions drawn from data analysis.
29. Describe ways in which computing enables innovation.
30. Explain how algorithms are used to produce artificial intelligence (AI)
31. Discuss the ways in which innovations enabled by computing affect communications and problem-solving.
32. Analyze how computing influences and is influenced by the cultures for which they are designed and the cultures in which they are used.
33. Analyze how social and economic values influence the design and development of computing innovations.
34. Discuss issues of equity, access, and power in the context of computing resources.
35. Communicate legal and ethical concerns raised by computational innovations.
36. Discuss privacy and security concerns related to computational innovations.
37. Explain the positive and negative effects of technological innovations on human culture.

Introduction to Database Design 110211

This course provides an overview of database and database management system concepts, internal design models, normalization, network data models, development tools, and applications. Students spend at least 20 hours of programming and applying learned concepts through programming. (Programming is defined, by the K-12 CS Framework, as the craft of analyzing problems and designing, writing, testing, and maintaining programs to solve them.) Participation in Kentucky Technology Student Association or SkillsUSA will greatly enhance instruction.

Recommended Grade Level: 9 – 12

Recommended Credit: 1

Students will:

1. Define a database and its uses.
2. Describe the difference between traditional files and databases.
3. Define a database management system (DBMS) and describe the services a DBMS provides to its users.
4. Identify and describe the main features of hierarchical, network, and relational database models.
5. Demonstrate an understanding of the difference between logical and physical design.
6. Model a realistic business application using a technology-independent data model.
7. Design and implement a database using the relational model, with emphasis on data integrity and security.
8. Define and use the normalization process to further refine the relational table definitions.
9. Demonstrate an understanding of the database administration function.
10. Define and be able to use data definition language, data manipulation language, and instructions that apply relational algebra.
11. Demonstrate an understanding of distributed database systems.
12. Evaluate and select an appropriate DBMS for a given application.

Introduction to Digital Game Graphics 113601

This course will focus on creating games using code, animation, and an introduction to 3D design software utilized in the industry. Students spend at least 20 hours of programming and applying learned concepts through programming. (Programming is defined, by the K-12 CS Framework, as the craft of analyzing problems and designing, writing, testing, and maintaining programs to solve them.) Participation in Kentucky Technology Student Association or SkillsUSA will greatly enhance instruction.

Recommended Grade Level: 9 – 12

Recommended Credit: 1

Students will:

1. Identify the target audience of a game.
2. Explain impact of “feature creep” on production.
3. Explain the interdependence of team members between artistic, technical and production disciplines.
4. Explain the purpose of prototyping.
5. Outline in detail the process of developing a game from concept to delivery and support.
6. Describe each step of the production process.
7. Explain how the project is going to be managed according to a milestone plan.
8. Explain the various types of collaboration tools.
9. Utilize the production pipeline in the development of a game.
10. Explain the value of version control.
11. Explain the purpose of vertical slice.
12. Demonstrate version control for example, Node Version Manager (NVM).
13. Demonstrate good quality assurance practices.
14. Conceptualize and illustrate original game characters and assets.
15. Utilize illustration to create assets.
16. Establish a standard for world scale.
17. Create a storyboard for planning animation.
18. Change an object’s state or position over time.
19. Establish an object’s relative speed.
20. Simulate a naturally occurring or mechanical cycle such as walking.
21. Apply animation to game assets.
22. Describe the role of typography.
23. Evaluate the use of layout and composition.
24. Explain color theory.
25. Describe the principles of animation.
26. Describe the role of perspective.
27. Demonstrate 1- and 2-point perspectives.
28. Draw a proportionally correct figure.
29. Describe the characteristics and purposes of 2D, 2.5D, and 3D art.
30. Recognize the importance of and implement continuity of art style.
31. Describe environments within a game.

32. Compare the process of creating an interior vs. exterior environment.
33. Identify components in an environment.
34. Generate terrains for a specific environment.
35. Create hard surface assets.
36. Create an environment.
37. Develop organics for a specific environment.
38. Describe archetypes of characters.
39. Explain character personalities and stereotypes.
40. Compart and contrast methods to design characters.
41. Describe the character’s evolution throughout the game.
42. Examine importance of non-player characters (NPC).
43. Construct character(s) for a game.
44. Differentiate between syntax and semantics.
45. Incorporate primitive data types.
46. Utilize arrays to store a list of primitive data types.
47. Demonstrate input from different sources.
48. Construct and register a callback function.
49. Compare and contrast constants and variables.
50. Select and implement conditional control.
51. Implement functions.
52. Select and implement iterations (loops, recursion,).
53. Recognize and implement sequential control.
54. Test and debug programs.
55. Design and implement user-defined data types.
56. Demonstrate output to different destinations.
57. Practice object-oriented programming (OPP).
58. Identify expected input and output.
59. Utilize basic steps in algorithmic problem-solving.
60. Discuss top-down versus bottom-up development.
61. Generate test cases and expected results.
62. Apply simple data structures.
63. Explain how algorithms are used to produce artificial intelligence (AI).

Introduction to Networking Concepts (non-vendor) 110901

This course introduces technical-level concepts of non-vendor-specific networking including technologies, media, topologies, devices, management tools, and security. Provides the basics of how to manage, maintain, troubleshoot, install, operate, and configure basic network infrastructure. Students spend at least 20 hours of programming and applying learned concepts through programming. (Programming is defined, by the K- 12 CS Framework, as the craft of analyzing problems and designing, writing, testing, and maintaining programs to solve them.) Participation in Kentucky Technology Student Association or SkillsUSA will greatly enhance instruction.

Recommended Grade Level: 9 – 12

Recommended Credit: 1

Students will:

1. Differentiate between network protocols in terms of routing, addressing schemes, interoperability, and naming conventions.
2. Identify addressing format, schemes, and technologies; and required settings for connectivity including classful/classless address ranges, public/private addressing, and subnetting.
3. Identify the common ports associated with TCP (Transmission Control Protocol) and UDP (User Datagram Protocol).
4. Identify the basic standards of WAN and remote access technologies.
5. Categorize standard cable types and their properties.
6. Explain logical and physical network structures, topologies, and characteristics.
7. Identify the basic attributes, purposes, and functions of network components including wireless technologies.
8. Identify the functions of specialized network devices such as multilayer switches, load balancer, proxy, DNS servers, and CSU/DSU.
9. Plan and implement a basic wired and wireless network.
10. Explain the function of each layer of the OSI and TCP/IP models.
11. Identify types of configuration management documentation such as network diagrams, wiring schematics, and configurations.
12. Summarize and explain different methods and rationales for network performance.
13. Diagnose a network problem using a systemic approach identifying the appropriate tools, selecting an appropriate course of action to resolve the problem, and document the solution.
14. Select the appropriate hardware and software tools to test, scan, and analyze network connectivity and performance.
15. Identify and explain common methods to ensure network security including antivirus software, user authentication, and firewall setup.
16. Identify issues that affect physical and remote access device security.

Introduction to Programming 110201

This course focuses on the general writing and implementation of generic and atomized programs to drive operating systems. Instruction includes software design, languages, program writing, and troubleshooting. Students are introduced to fundamental programming concepts using an industry-specific or emerging programming language. Includes data types, control structures, simple data structures, error-handling, modular programming, information and file processing, and uniqueness of the language used in the course. Students spend at least 20 hours of programming and applying learned concepts through programming. (Programming is defined, by the K-12 CS Framework, as the craft of analyzing problems and designing, writing, testing, and maintaining programs to solve them.) Participation in Kentucky Technology Student Association or SkillsUSA will greatly enhance instruction.

Recommended Grade Level: 9 – 12

Recommended Credit: 1

Students will:

1. Demonstrate knowledge of the program development life cycle.
2. Design, develop, compile, debug, test, run, and document programs in the language studied.
3. Design and develop programs using operators and assignments.
4. Design and develop programs that properly use variable, constants, data types, and objects.
5. Design and develop programs that use sequence, selection, and repetition structures.
6. Design and develop programs that use simple data structures.
7. Design and develop programs that use effective error and exception handling.
8. Design and develop programs that implement user-defined methods and modular programming.
9. Design and develop programs that implement file processing.
10. Design and develop programs that implement fundamental features that are unique to the language studied.
11. Design and develop programs using object-oriented programming features, if applicable to the language studied.
12. Explain how algorithms are used to produce artificial intelligence (AI).
13. Evaluate and critique the effectiveness and efficiency of code written.

JAVA Programming I 110205

Java Programming I introduces students to fundamental programming concepts using the Java programming language. Topics include data types, control structures, simple data structures, error-handling, object-oriented programming, graphical user interfaces, and modular programming. Students spend at least 20 hours of programming and applying learned concepts through programming. (Programming is defined, by the K-12 CS Framework, as the craft of analyzing problems and designing, writing, testing, and maintaining programs to solve them.) Participation in Kentucky Technology Student Association or SkillsUSA will greatly enhance instruction.

Recommended Grade Level: 10 – 12

Recommended Credit: 1

Students will:

1. Design, develop, compile, debug, test, run, and document programs in the Java language using a software development kit.
2. Design and develop programs using operators and assignments.
3. Design and develop programs using primitive data types.
4. Design and develop programs using sequence, selection, and repetition structures.
5. Design and develop programs using single and multi-dimensional arrays.
6. Design and develop programs using effective error and exception handling.
7. Design and develop programs using object-oriented programming features, including defining classes, instantiating objects, and using arrays of objects.
8. Design and develop programs implementing user-defined methods and modular programming.
9. Design and develop programs using method overloading.
10. Design and develop programs using inheritance, encapsulation, and polymorphism.
11. Design and develop GUI interfaces for Java applications.
12. Evaluate and critique the effectiveness and efficiency of code.

JAVA Programming II 110206

Java Programming II provides students with an extensive overview of designing and developing advanced object-oriented applications using the Java programming language. Topics include input and output streams (file processing), polymorphism, inheritance, multithreading, recursion, mobile computing, and other advanced topics. Students spend at least 20 hours of programming and applying learned concepts through programming. (Programming is defined, by the K-12 CS Framework, as the craft of analyzing problems and designing, writing, testing, and maintaining programs to solve them.) Participation in Kentucky Technology Student Association or SkillsUSA will greatly enhance instruction.

Recommended Grade Level: 10 – 12

Recommended Credit: 1

Students will:

1. Design and develop programs that use advanced GUI components.
2. Design and develop programs that use input and output streams including character and binary streams.
3. Design and develop programs that use multithreading.
4. Design and develop programs that use polymorphism.
5. Design and develop programs that use inheritance.
6. Design and develop programs that use recursion.
7. Design and develop programs that introduce mobile application concepts.
8. Design and develop programs that incorporate other advanced features of Java programming.
9. Evaluate and critique the effectiveness and efficiency of code.

JavaScript 110809

This course provides students with an overview of the JavaScript scripting language. Includes coding, testing, and debugging JavaScript programs; using a variable, operators, and data types; creating dynamic web pages using JavaScript; controlling the behavior of forms, buttons, and text elements; and using control structures, pattern matching, objects, and application scripts. Students spend at least 20 hours of programming and applying learned concepts through programming. (Programming is defined, by the K-12 CS Framework, as the craft of analyzing problems and designing, writing, testing, and maintaining programs to solve them.) Participation in Kentucky Technology Student Association or SkillsUSA will greatly enhance instruction.

Recommended Grade Level: 10 – 12

Recommended Credit: 1

Students will:

1. Design, develop, compile, debug, test, run, and document programs in the JavaScript language.
2. Design and develop programs using operators and assignments.
3. Design and develop programs using a variety of data types.
4. Demonstrate the input and output processes in JavaScript.
5. Design and develop programs using sequence, selection, and repetition structures.
6. Demonstrate pattern matching using JavaScript.
7. Demonstrate JavaScript objects.
8. Demonstrate the ability to write JavaScript application scripts.
9. Evaluate and critique the effectiveness and efficiency of code.

LAN Switching and Wireless/Scaling Networks/Cisco III 110904

This course covers the architectures and considerations related to designing, securing, operating, and troubleshooting enterprise networks. It covers wide area network (WAN) technologies and quality of service (QoS) mechanisms used for secure remote access along with the introduction of software-defined networking, virtualization, and automation concepts that support the digitalization of networks. Students will learn how to configure routers and switches for advanced functionality. Students spend at least 20 hours of programming and applying learned concepts through programming. (Programming is defined, by the K-12 CS Framework, as the craft of analyzing problems and designing, writing, testing, and maintaining programs to solve them.) Participation in Kentucky Technology Student Association or SkillsUSA will greatly enhance instruction.

Recommended Grade Level: 10 – 12

Recommended Credit: 1

Students will:

1. Configure and troubleshoot DHCP and DNS operations for IPv4 and IPv6.
2. Describe the operations and benefits of the Spanning Tree Protocol (STP).
3. Configure and troubleshoot STP operations.
4. Describe the operations and benefits of link aggregation and Cisco VLAN Trunk Protocol (VTP).
5. Configure and troubleshoot VTP, STP, and RSTP.
6. Configure and troubleshoot basic operations of routers in a complex routed network for IPv4 and IPv6.
7. Configure Open Shortest Path First (OSPF) protocol (single-area OSPF and multi-area OSPF).
8. Configure Enhanced Interior Gateway Routing Protocol (EIGRP).
9. Configure and troubleshoot advanced operation of routers and implement RIP, OSPF, and EIGRP routing protocols for IPv4 and IPv6.
10. Configure and manage Cisco IOS Software licensing and configuration files.
11. Mitigate threats and enhance network security using access control lists and security best practices.
12. Develop critical thinking and problem-solving skills using real equipment and Cisco Packet Tracer.
13. Understand virtualization, SDN, and how APIs and configuration management tools enable network automation.

Leadership Dynamics – Information Technology 110399

This course is designed to assist students with developing skills needed to be successful leaders and responsible members of society. This student will develop personal attributes and social skills. Emphasis will be placed on interpersonal skills, team building, communication, personal development, and leadership. This course will include opportunities for students to apply their knowledge. Students spend at least 20 hours of programming and applying learned concepts through programming. (Programming is defined, by the K-12 CS Framework, as the craft of analyzing problems and designing, writing, testing, and maintaining programs to solve them.) Participation in Kentucky Technology Student Association or SkillsUSA will greatly enhance instruction.

Recommended Grade Level: 9 – 12

Recommended Credit: .5 – 1

Students will:

1. Investigate types of leadership and determine personal style.
2. Compare and contrast positive and negative characteristics of leaders.
3. Identify the role of leadership in the global society.
4. Assess the role that qualified leaders have on the success of organizations.
5. Explain how cultural and social diversity and equity impact leadership skills.
6. Identify and explain the importance of team membership skills for individuals and groups.
7. Develop interpersonal skills for resolving conflicts that occur in the home, school, community and workplace.
8. Demonstrate verbal and nonverbal communication skills needed for personal and leadership roles.
9. Make informed decisions using the decision-making process.
10. Demonstrate appropriate parliamentary procedure skills used in meetings.
11. Analyze leadership opportunities available in school and community.
12. Describe how ethical and social behaviors affect individuals.
13. Develop personal goals.
14. Demonstrate appropriate business, progression and social etiquette.
15. Analyze the role self-management has on the use of time and stress.

Management of Support Services 110302

Digitally organize the information technology and information and support services milestones achieved by the student that is reflective of their industry certification readiness, understanding the cost of doing business, and preparation of technical and behavioral job performances such as interviews. The course also focuses on employability skills to include: a professional digital portfolio that emphasizes critical milestones focusing on entry-level information technology technical and employability skills. Students spend at least 20 hours of programming and applying learned concepts through programming. (Programming is defined, by the K-12 CS Framework, as the craft of analyzing problems and designing, writing, testing, and maintaining programs to solve them.) Participation in Kentucky Technology Student Association or SkillsUSA will greatly enhance instruction.

Recommended Grade Level: 11 – 12

Recommended Credit: 1

Students will:

1. Determine purpose and goals using a project management method.
2. Define, Design, Develop, Deploy, Reflect, Redesign, and Present utilizing presentation software visualizing the process.
3. Determine roles, tasks, calendars.
4. Utilize Software packages for project management (MS Project, Excel, Visio, Dread-Spark, Prezi).
5. Utilize and define appropriate terminology.
6. Present information in a technical report.
7. Publish information presented to an advisory board member.
8. Identify potential employment barriers for non-traditional groups and ways to overcome the barrier.
9. Research potential barriers to placing information in a spreadsheet.
10. Present information to school principals and peers.
11. Synthesize the information collecting producing a product that could help overcome the barrier with non-traditional groups.

Microsoft Client/Server Configuration 110913

This course covers the installation and configuration of Microsoft Windows client and server operating systems. Students spend at least 20 hours of programming and applying learned concepts through programming. (Programming is defined, by the K-12 CS Framework, as the craft of analyzing problems and designing, writing, testing, and maintaining programs to solve them.) Participation in Kentucky Technology Student Association or SkillsUSA will greatly enhance instruction.

Recommended Grade Level: 10 – 12

Recommended Credit: 1

Students will:

1. Evaluate hardware compatibility for installation.
2. Install and upgrade the Windows client operating system and migrate user data.
3. Manage file systems, partitions, disks and devices.
4. Install, configure, and control access to applications including desktop applications and Windows Store apps.
5. Configure Client Hyper-V.
6. Configure Transmission Control Protocol/Internet Protocol (TCP/IP) settings and network security settings.
7. Configure Remote Management and remote connections.
8. Configure local and shared access for files and folders.
9. Configure, secure, and manage mobile computing.
10. Monitor and optimize operating system performance and resource usage.
11. Implement disaster protection, including backup and file recovery options.
12. Create and manage user accounts and groups.
13. Configure printing and print services.
14. Configure and troubleshoot the boot process and System Recovery options.

Network Fundamentals/Cisco I 110902

This course introduces the architecture, structure, functions, components, and models of the Internet and other computer networks. Students are introduced to the principles and structure of IP addressing and the fundamentals of Ethernet concepts, media, and operations. Helps students to be able to build simple LANs, perform basic configurations for routers and switches, and implement IP addressing schemes. Students spend at least 20 hours of programming and applying learned concepts through programming. (Programming is defined, by the K-12 CS Framework, as the craft of analyzing problems and designing, writing, testing, and maintaining programs to solve them.) Participation in Kentucky Technology Student Association or SkillsUSA will greatly enhance instruction.

Recommended Grade Level: 9 – 12

Recommended Credit: 1

Students will:

1. Describe the devices and services used to support communications in data networks and the Internet.
2. Describe the role of protocol layers in data networks.
3. Describe the importance of addressing and naming schemes at various layers of data networks in IPv4 and IPv6 environments.
4. Design, calculate and apply subnet masks and addresses to fulfill given requirements in IPv4 and IPv6 networks.
5. Explain fundamental Ethernet concepts such as media, services, and operations.
6. Design a simple Ethernet network using routers, switches, cables, connectors and other hardware.
7. Demonstrate the Cisco command-line interface (CLI) commands to perform basic router and switch configurations.
8. Utilize common network utilities to verify small network operations and analyze data traffic.

Network Hardware Installation and Troubleshooting 110906

This course is designed to provide students with the knowledge and skills necessary to design, install, configure, and troubleshoot cabling systems and equipment used to connect a local area network. Students spend at least 20 hours of programming and applying learned concepts through programming. (Programming is defined, by the K-12 CS Framework, as the craft of analyzing problems and designing, writing, testing, and maintaining programs to solve them.) Participation in Kentucky Technology Student Association or SkillsUSA will greatly enhance instruction.

Recommended Grade Level: 10 – 12

Recommended Credit: 1

Students will:

1. Design a basic network layout using copper and/or fiber optic cabling systems.
2. Terminate, test, and troubleshoot copper wire systems.
3. Install and configure network interface cards and connection equipment.
4. Use industrial standard testing and certification equipment.

Object-Oriented Programming I 110220

This course introduces students to fundamental programming concepts using an Object-Oriented Programming language(s). Teachers select the programming language that is most appropriate for their students. Topics include data types, control structures, simple data structures, arrays, GUI, modular programming and error-handling. Students spend at least 20 hours of programming and applying learned concepts through programming. (Programming is defined, by the K-12 CS Framework, as the craft of analyzing problems and designing, writing, testing, and maintaining programs to solve them.)

Recommended Grade Level: 10 – 12

Recommended Credit: 1

Students will:

1. Design, develop, compile, debug, test, run and document programs.
2. Demonstrate knowledge of the program development life cycle.
3. Design and develop programs using operators and assignments.
4. Design and develop programs using primitive data types.
5. Design and develop programs using a variety of data types.
6. Design and develop programs using sequences, selection, and repetition structures.
7. Design and develop programs using single and multi-dimensional arrays.
8. Design and develop programs using arrays, lists and tuples.
9. Utilize arrays
10. Design and develop programs using effective error and exception handling.
11. Design and develop programs using object-oriented programming features, including defining classes, instantiating objects, and using arrays of objects.
12. Design and develop programs implementing user-defined methods and modular programming.
13. Explain how algorithms are used to produce artificial intelligence (AI).
14. Design and develop programs using method overloading.
15. Design and develop programs using inheritance, encapsulation, and polymorphism.
16. Design and develop programs using simple GUI components.
17. Evaluate and critique the effectiveness and efficiency of code.

Object-Oriented Programming II 110221

This course provides students with an extensive overview of designing and developing advanced object-oriented applications. Teachers select the programming language(s) most appropriate for their students. Topics include input and output streams (file processing), polymorphism, inheritance, multithreading, recursion, mobile computing, and other advanced topics. Students spend at least 20 hours of programming and applying learned concepts through programming. (Programming is defined, by the K-12 CS Framework, as the craft of analyzing problems and designing, writing, testing, and maintaining programs to solve them.) Participation in Kentucky Technology Student Association or SkillsUSA will greatly enhance instruction.

Recommended Grade Level: 10 – 12

Recommended Credit: 1

Students will:

1. Design and develop programs that use advanced GUI Components.
2. Design and develop programs that use input and output streams including character and binary streams.
3. Demonstrate knowledge of advanced concepts and associated definitions.
4. Design and code applications using advanced data types and structures.
5. Design and develop programs that use concurrency.
6. Design, develop, compile, debug, test, run and document advanced programs in the language.
7. Design and develop programs using polymorphism, inheritance, and overloading.
8. Design and develop programs that incorporate other advanced features.
9. Examine and evaluate the strengths and weaknesses of the language(s).
10. Demonstrate error-checking and error handling.
11. Implement input validation and processing.
12. Evaluate and critique the effectiveness and efficiency of code.

Productivity Software 110204

This course utilizes current word processing, spreadsheet, databases, and presentation application software to solve common technology and business problems. It covers the basic features of each software application. Students spend at least 20 hours of programming and applying learned concepts through programming. (Programming is defined, by the K-12 CS Framework, as the craft of analyzing problems and designing, writing, testing, and maintaining programs to solve them.) Participation in Kentucky Technology Student Association or SkillsUSA will greatly enhance instruction.

Recommended Grade Level: 10 – 12

Recommended Credit: 1

Students will:

1. Use a productivity software package to create, edit, print, and save documents.
2. Use productivity tools such as spelling and grammar.
3. Apply formatting features such as font, color, margins, headers, footers.
4. Use tools such as cut, copy, and paste within a document and between documents.
5. Create HTML file formats for web publishing.
6. Create new documents using templates and wizards.
7. Use a word processing program to insert and use table features.
8. Use a word processing program to insert and use table column features.
9. Insert pictures and Clipart into word processing documents.
10. Use a spreadsheet package to create common business reports and budgets.
11. Use mathematical formulas and common statistical, date, financial, and logical functions.
12. Make formatting changes to a worksheet including column width, row height, cell, and table formatting.
13. Use autofill to copy and paste formulas and repeat patterns.
14. Create effective charts, including bar, line, and pie charts, to accompany business reports.
15. Use a relational database management program to create tables, queries, forms, reports, and labels.
16. Use query features to extract information from a database using simple and compound conditions.
17. Use relationship feature to join tables in a database and obtain information from multiple tables.
18. Plan and create an electronic slide show presentation using a presentation software package.
19. Use timing, transition, and animation features to enhance a presentation.

Project-Based Programming 110226

This project-based learning course engages those students with an entrepreneurial spirit that are interested in programming and in finding solutions to existing problems through the creation of applications. In this course, students will create projects that require computer science fundamentals and extensive research for successful completion. Students will work either solo or in a team to execute a project decided upon by the student(s). Students must learn and demonstrate proficiency in time management, scope, research, computer science, and teamwork to be successful in this course. Finally, students will engage in leadership skills by being held accountable for the completion of their tasks or project. The teacher will act more as a facilitator in this course and is highly encouraged to create his/her own project to demonstrate teacher “buy-in” to students. Students spend at least 20 hours of programming and applying learned concepts through programming. (Programming is defined, by the K-12 CS Framework, as the craft of analyzing problems and designing, writing, testing, and maintaining programs to solve them.) Participation in Kentucky Technology Student Association or SkillsUSA will greatly enhance instruction.

Recommended Grade Level: 10 – 12

Recommended Credit: 1

Students will:

1. Manage and modify the scope to ensure success.
2. Apply time management skills to effectively complete projects to specifications.
3. Use programming language of choice to develop a project of choice.
4. Demonstrate continuous improvement through feedback from one-on-one meetings each week and regular project updates with the facilitator.
5. Use appropriate and effective research skills.
6. Demonstrate proficiency in computer science fundamentals and design patterns.
7. Architect a project for the semester.
8. Demonstrate the use of version control through Git.
9. Demonstrate manipulation of search terms in Google to obtain valid and useful results.

Routing Protocols and Concepts/Cisco II 110903

This course focuses on switching technologies and router operations that support small-to-medium business networks, including wireless local area networks (WLAN) and security concepts. Students perform basic network configuration and troubleshooting, identify and mitigate LAN security threats, and configure and secure a basic WLAN. Students spend at least 20 hours of programming and applying learned concepts through programming. (Programming is defined, by the K-12 CS Framework, as the craft of analyzing problems and designing, writing, testing, and maintaining programs to solve them.) Participation in Kentucky Technology Student Association or SkillsUSA will greatly enhance instruction.

Recommended Grade Level: 10 – 12

Recommended Credit: 1

Students will:

1. Describe basic switching concepts and the operation of Cisco switches.
2. Describe enhanced switching technologies such as VLANs, VLAN Trunking Protocol (VTP), Rapid Spanning Tree Protocol (RSTP), Per VLAN Spanning Tree Protocol (PVSTP), and 802.1q.
3. Troubleshoot basic operations of a small switched network.
4. Describe the purpose, nature, and operations of a router, routing tables, and the route lookup process.
5. Configure and troubleshoot static routing and default routing.
6. Describe how VLANs create logically separate networks and how routing occurs between them.
7. Describe dynamic routing protocols, distance vector routing protocols, and link-state routing protocols.
8. Configure and troubleshoot basic operations of routers in a small routed network using Routing Information Protocol (RIPv1 and RIPv2).
9. Configure and troubleshoot basic operations of routers in a small routed network using Configure Open Shortest Path First (OSPF) protocol (single-area OSPF).
10. Configure and troubleshoot VLANs, Wireless LANs and inter-VLAN routing.
11. Describe the purpose and types of access control lists (ACLs).
12. Prepare, monitor, and troubleshoot access control lists (ACLs) for IPv4 and IPv6.
13. Describe the operations and benefits of Dynamic Host Configuration Protocol (DHCP) and Domain Name System (DNS) for IPv4 and IPv6.
14. Describe the operations and benefits of Network Address Translation (NAT).
15. Configure and troubleshoot Network Address Translation (NAT) operations.
16. Configure and troubleshoot redundancy on a switched network using STP and Ether Channel.
17. Develop critical thinking and problem-solving skills using real equipment and Cisco Packet Tracer.
18. Explain how to support available and reliable networks using dynamic addressing and first-hop redundancy protocols.
19. Configure port security to mitigate MAC address table attacks.

Security Fundamentals 110912

Security Fundamentals introduces basic computer and network security concepts and methodologies. Covers principles of security; compliance and operational security; threats and vulnerabilities; network security; application, data, and host security; access control and identity management; and cryptography. Students spend at least 20 hours of programming and applying learned concepts through programming. (Programming is defined, by the K-12 CS Framework, as the craft of analyzing problems and designing, writing, testing, and maintaining programs to solve them.) Participation in Kentucky Technology Student Association or SkillsUSA will greatly enhance instruction.

Recommended Grade Level: 10 – 12

Recommended Credit: 1

Students will:

1. Explain basic security concepts.
2. Identify and explain the appropriate use of security tools to facilitate security.
3. Evaluate current security issues related to computer and network systems.
4. Evaluate and select appropriate incident response procedures, disaster recovery, and risk identification techniques to ensure business continuity.
5. Differentiate various malware and systems security threats against computers and networks.
6. Explain the vulnerabilities and mitigations associated with computers and network devices.
7. Explain the proper use of common tools for carrying out vulnerability assessments.
8. Identify and describe potential application and data vulnerabilities, including buffer overflow, DLL injection, and SQL injection.
9. Explain how host firewalls, malware protection, and updates are important to application and data security.
10. Describe the importance of user accounts and associated permissions.
11. Compare and discuss logical and physical access control security methods.
12. Explain authentication models and identify components of each model.
13. Summarize and explain general cryptography concepts.
14. Demonstrate public and private key pairs for digital signing and encryption/decryption.

Special Topics - Computer Science 110752

Special Topics courses may be utilized, with justification for the course and course objectives, upon approval by the Computer Science Consultant related to career major. Students spend at least 20 hours of programming and applying learned concepts through programming. (Programming is defined, by the K-12 CS Framework, as the craft of analyzing problems and designing, writing, testing, and maintaining programs to solve them.)

Recommended Grade Level: 11 – 12

Recommended Credit: 1

Students will:

1. Complete tasks defined by the teacher and approved by the Computer Science Consultant in the Office of Career and Technical Education.

Special Topics - Information Support and Services 110152

Special Topics courses may be utilized, with justification for the course and course objectives, upon approval by the Computer Science Consultant related to career major. Students spend at least 20 hours of programming and applying learned concepts through programming. (Programming is defined, by the K-12 CS Framework, as the craft of analyzing problems and designing, writing, testing, and maintaining programs to solve them.)

Recommended Grade Level: 11 – 12

Recommended Credit: 1

Students will:

1. Complete tasks defined by the teacher and approved by the Computer Science Consultant in the Office of Career and Technical Education.

Special Topics - Networking 110952

Special Topics courses may be utilized, with justification for the course and course objectives, upon approval by the Computer Science Consultant related to career major. Students spend at least 20 hours of programming and applying learned concepts through programming. (Programming is defined, by the K-12 CS Framework, as the craft of analyzing problems and designing, writing, testing, and maintaining programs to solve them.)

Recommended Grade Level: 11 – 12

Recommended Credit: 1

Students will:

1. Complete tasks defined by the teacher and approved by the Computer Science Consultant in the Office of Career and Technical Education.

Special Topics - Programming 110252

Special Topics courses may be utilized, with justification for the course and course objectives, upon approval by the Computer Science Consultant related to career major. Students spend at least 20 hours of programming and applying learned concepts through programming. (Programming is defined, by the K-12 CS Framework, as the craft of analyzing problems and designing, writing, testing, and maintaining programs to solve them.)

Recommended Grade Level: 11 – 12

Recommended Credit: 1

Students will:

1. Complete tasks defined by the teacher and approved by the Computer Science Consultant in the Office of Career and Technical Education.

Special Topics - Web Development/Administration 110852

Special Topics courses may be utilized, with justification for the course and course objectives, upon approval by the Information Technology Consultant related to career major. Students spend at least 20 hours of programming and applying learned concepts through programming. (Programming is defined, by the K-12 CS Framework, as the craft of analyzing problems and designing, writing, testing, and maintaining programs to solve them.)

Recommended Grade Level: 11 – 12

Recommended Credit: 1

Students will:

1. Complete tasks defined by the teacher and approved by the Computer Science Consultant in the Office of Career and Technical Education.

Web Page Development 110801

This course introduces web pages through the use of HTML and CSS. Students use text and/or web editors to create web documents with various formats and page layouts, multimedia, tables, and forms. Instruction emphasizes W3C web design and accessibility standards. Students spend at least 20 hours of programming and applying learned concepts through programming. (Programming is defined, by the K-12 CS Framework, as the craft of analyzing problems and designing, writing, testing, and maintaining programs to solve them.) Participation in Kentucky Technology Student Association or SkillsUSA will greatly enhance instruction.

Recommended Grade Level: 10 – 12

Recommended Credit: 1

Students will:

1. Plan the layout of a website.
2. Use HTML (Hypertext Markup Language).
3. Use CSS (Cascading Style Sheets).
4. Create lists and tables in an organization's content.
5. Use HTML and CSS in page layout.
6. Create web forms.
7. Use multimedia in the creation of a website (such as images, sound, and video).
8. Publish web pages to a website.

Web Site Design and Production 110804

This course introduces website production processes with particular emphasis on a design involving layout, navigation, interactivity, and using web production software. Students spend at least 20 hours of programming and applying learned concepts through programming. (Programming is defined, by the K-12 CS Framework, as the craft of analyzing problems and designing, writing, testing, and maintaining programs to solve them.) Participation in Kentucky Technology Student Association or SkillsUSA will greatly enhance instruction.

Recommended Grade Level: 10 – 12

Recommended Credit: 1

Students will:

1. Utilize principles of graphic and content creation for online media.
2. Use fundamental online graphic design principles including appropriate interactivity, content sensitive navigation schemes and user interface criteria.
3. Select task-appropriate software tools.
4. Utilize website accessibility.
5. Utilize website implementation and hosting.
6. Edit and enhance digital video images using state-of-the-art software.