Code.org Computing Foundations for a Digital Age: Indiana (AI Foundations) Syllabus and Overview

Overview

Computing Foundations for a Digital Age: Indiana (AI Foundations) introduces students to the core principles of computer science and the growing role of artificial intelligence in everyday life. Through hands-on programming, visual problem solving, and critical analysis, students explore how computing systems work, how data powers AI, and how intelligent tools make decisions. They learn to write Python programs, analyze the structure of the Internet, investigate cybersecurity risks, and interpret data through visualizations and models. Throughout the course, students examine how AI systems impact individuals and communities, and build the skills to use computing and AI to solve meaningful problems in their world.

Learning Outcomes

By the end of the CFDA: Indiana (AIF) curriculum, students will be able to:

- Explain how generative AI models work and evaluate their outputs based on accuracy, clarity, and ethical considerations.
- Design and implement programs in Python using functions, loops, conditionals, and abstraction to solve structured problems.
- Use computational thinking strategies such as decomposition and pattern recognition to solve problems and analyze AI behavior.
- Describe how hardware and software components interact to input, store, process, and output information in computing systems.
- Explain how data moves across networks and the Internet, and analyze how AI improves reliability, accessibility, and user experience online.
- Identify and evaluate common cybersecurity threats and apply strategies like encryption and authentication to protect information.
- Analyze the social, ethical, and privacy implications of computing innovations and AI systems, especially regarding data use and decision-making.
- Interpret how AI systems use data to recognize patterns, make predictions, and influence real-world outcomes.
- Reflect on the role of personal identity and values in shaping how students use, critique, and create with computing and AI tools.

Prerequisites

CFDA: Indiana (AIF) is designed for high school students at any level of computer science experience. It is accessible to beginners with no prior programming knowledge, and it also provides opportunities for students with some background in computing to deepen their understanding of key topics.

In Indiana, this course maps to 4565: Computing Foundations for a Digital Age which is a high-school course that can be offered in middle school. To learn more, see the NLPS website here: https://www.in.gov/che/cte/career-pathways-programs-of-study/

This curriculum can serve as an introductory course or as a bridge to more advanced studies, such as AP Computer Science A (CSA).

No specific prior coursework is required, and all necessary concepts are introduced and reinforced through guided activities, collaborative learning, and hands-on projects.

Our Vision

Code.org's vision is that every student in every school should have the opportunity to learn computer science (code.org/about). Our curriculum is designed so that an empowered teacher can lead a diverse group of students through experiences that are supportive, equitable, engaging, and lead to valuable learning (code.org/educate/curriculum/values).

Historically, this vision has contrasted sharply with reality. Until recently, most schools did not offer computer science, and schools that did offer computer science did not have enrollment that matched the demographics of their school population. Additionally, many students found these classes unengaging, intimidating, or disconnected from their lived experiences with technology. Thanks to efforts by many organizations and individuals, this world is beginning to change: many more schools now offer computer science courses; the diversity of schools enrolled in those courses is increasing; and more engaging, relevant, and equitable pedagogy has become the established norm. Even so, there is much work still to be done. This course is designed to continue this momentum as the collective CS education community moves towards this vision of an equitable CS education system.

How We Support Our Vision

Many aspects of CFDA: Indiana (AIF) are designed to bring about the eventual change we aim to see more broadly in CS education. Some of the most significant features are listed below.

- **Free and Open:** We make our curriculum, videos, and tools free and open for anyone to use.
- **Modular Design for Flexibility:** CFDA: Indiana (AIF) is designed to be modular, allowing teachers to adjust the sequence of units based on their classroom context and students' needs.
- **Student-Centered Pedagogy:** We aim to develop curriculum that ensures students' identities, backgrounds, and experiences are recognized and celebrated as assets. Lessons are designed to reflect the cultural and personal interests of students, fostering a sense of belonging and engagement. Students are encouraged to apply computing concepts to real-world problems relevant to their lives and communities, promoting meaningful connections between technology and society.
- **Collaborative, Supportive Learning Environments:** CFDA: Indiana (AIF) emphasizes inclusive collaboration through practices like pair programming, group activities, and peer feedback. These collaborative experiences help students build empathy, communication skills, and trust, which are essential for working effectively in diverse teams. Reflection activities encourage students to critically engage with their learning process, building persistence and resilience.
- **Ethical and Societal Awareness:** Every unit integrates discussions on ethics, equity, and the societal impact of technology. Students explore topics such as algorithmic bias, data privacy, and accessibility,

empowering them to think critically about how computing affects individuals and communities. Through human-centered design principles, students learn to create solutions that meet the needs of diverse users, ensuring their work has meaningful social impact.

- **Encouraging Real-World Problem Solving and Creativity:** Hands-on projects allow students to apply computational thinking to solve real-world challenges, such as developing AI tools for social good or addressing the digital divide. These projects give students opportunities for creative expression and empower them to see computing as more than just a technical field–it is a tool for shaping the world around them.
- **Cultivating Key Dispositions for Lifelong Learning:** Throughout the curriculum, students develop dispositions such as curiosity, persistence, resourcefulness, and a sense of belonging. These qualities are nurtured through project-based learning and iterative feedback, preparing students to confidently engage with technology in future studies, careers, and civic life.

Materials and Resources

The curriculum provides a comprehensive set of resources for the teacher, including detailed daily lesson plans, engaging activities and projects, formative and summative assessments, computing tools that are designed for learning specific concepts, and the Python Lab programming environment. These resources have been specifically curated to provide a unified experience for teachers and students. Together, these resources allow the teacher to act as a facilitator and coach for their students when addressing unfamiliar material. When the teacher acts as the primary source of information, generous support is provided.

All resources can be accessed free of charge at https://studio.code.org/courses/computing-foundations-for-a-digital-age-2025.

A full standard alignment of the CFDA: Indiana course to the Indiana 4565 course standards can be found at https://docs.google.com/spreadsheets/d/1165dQAhsK1vYOc86nrCJA9Ey0IQaH9qS8RNyeoxcc1c/edit?usp=sharing

The following resources and information can be found in each lesson plan:

- Instructions and teaching tips for conducting the lesson
- Activity guides and handouts for students
- Lesson slide decks
- Formative and summative assessments
- Answer keys, exemplars, and rubrics

Code.org Computing Foundations for a Digital Age: Indiana (AI Foundations) Syllabus and Overview

Getting Verified

A verified teacher is a Code.org teacher account that we can prove belongs to a teacher. By becoming verified, you will get access to answer keys and project exemplars.

You can become a verified teacher by:

- **Attending Code.org Professional Development for our AIF or CFDA: Indiana (AIF) curriculum.** This process should happen automatically once you have attended professional development.
- **Being manually verified as an actual teacher.** If you have not gone through our professional development, you can apply to become verified by filling out the form at https://form.jotform.com/240863841362155. We manually review each response - this process takes on average one business day. You will receive an email once you are verified successfully. If you don't hear back from us after a few business days, contact us at verification@code.org.

Technical Requirements

The curriculum requires and assumes a 1:1 computer lab or a setup such that each student in the class has access to an Internet-connected computer every day in class. All curriculum tools and resources are available online. Tablets are not currently supported. For more details on the technical requirements, please visit code.org/educate/it.

The curriculum does not require you to download any programs on your or your students' computers. The curriculum only requires computers that have browser access to Code.org from Chrome, Edge, Firefox, or Safari. Internet Explorer is not supported.

While the curriculum features many unplugged activities designed to be completed without a computer, daily access to a computer is essential for every student. CFDA: Indiana (AIF) is developed to be completed within the classroom – no homework or after-hours computer access is assumed.

Additional Materials and Supplies

One potentially significant cost to consider is printing. Many lessons have handouts that are designed to guide students through activities. While it is not required that all of these handouts be printed, many were designed to be printed, and we highly recommend their use.

Beyond printing, some lessons call for typical classroom supplies and manipulatives such as:

- poster paper
- markers or colored pencils
- sticky notes

Suggested substitutes can be found in individual lesson plans.

Code.org Computing Foundations for a Digital Age: Indiana (AI Foundations) Syllabus and Overview

Course Structure

The entire code.org CFDA: Indiana (AIF) course is organized into two semesters, each comprising modular units. This flexible structure allows teachers to tailor the curriculum to meet their classroom needs while ensuring a comprehensive exploration of foundational computer science and AI concepts.

For Indiana teachers using the CFDA: Indiana (AIF) course, the **first semester** fully aligns with the approved state course 4565: Computing Foundations for a Digital Age and is a high-quality curricular option for the course via the IDOE that meets the course competencies.
https://www.in.gov/doe/students/high-quality-curricular-materials-advisory-lists/#Technology__Computer_Science_

- **Semester 1:** Focuses on introducing students to core CS principles, the societal and ethical impacts of technology, and an overview of AI. Topics include generative AI, programming fundamentals, computer systems, networks, cybersecurity, and data science. These foundational units provide students with the knowledge and skills needed to engage critically with technology.
- **Semester 2** (*Coming Soon!*)**:** Emphasizes hands-on Python programming and the development of various AI applications. Building on the programming fundamentals from Semester 1, students dive deeper into creating real-world projects. Topics include machine learning concepts and developing AI-driven tools. Students apply Python programming skills to solve complex problems and implement AI solutions, all while exploring the ethical and societal implications of their work.

Each unit in the curriculum integrates computational thinking, ethical discussions, and career exploration, providing students with practical experience and a broader understanding of how technology shapes the world. The modular design ensures that teachers can adapt the course content to fit a variety of classroom settings, schedules, and student interests.

Code.org Computing Foundations for a Digital Age: Indiana (AI Foundations) Syllabus and Overview

Curriculum at a Glance: Semester 1

Unit	Topics Covered
Unit 1: Problem Solving with AI	AI as an Assistant, AI Limitations, Bias and Hallucinations, AI for Creativity, AI in Decision-Making
Unit 2: Foundations of AI Programming	Variables, Loops, Conditionals, Functions, Debugging, Ethical Computing
Unit 3: AI and the Systems That Power It	Hardware, Software, Data Processing, IoT, Emerging Technologies
Unit 4: The Fabric of the Internet and AI	Data Transmission, IP Addresses, Cybersecurity, Digital Divide
Unit 5: Cybersecurity and Global Impacts	Privacy, Security Risks, Ethical Computing, Encryption
Unit 6: Insights from Data and AI	Data Questions, Data Cleaning and Analysis, Visualizations, Data Stories, Ethical Considerations

Unit 1: Problem Solving with AI

wk	
1	Talking to Machines Beyond Words The AI's Brain Smart or Just Predictable? Uncovering Contradictions
2	Understanding Bias AI's Wild Imagination Debugging and Refining Outputs AI as a Co-Creator Remixing, Creativity, and Originality
3	Ownership, Ethics, and Creativity AI's Role in Society AI Time Capsule Project

Unit 2: Foundations of AI Programming

wk	
1	Computing Careers Computational Thinking Designing Algorithms Variables and Data Flow Working with Objects
2	Functions Functions with Parameters Introduction to Loops Conditional Statements Algorithms IRL
3	Decomposition Two-Way Selection Pixels of Me Project

Unit 3: AI and the Systems That Power It

wk	
1	Introduction to Computer Systems Hardware Troubleshooting Hardware Software and Operating Systems Troubleshooting Software
2	Build a Computer System Data in Computer Systems Processing Data in a Computer System How AI Uses Data IoT and Emerging Technologies
3	User Testing Just Because We Can, Should We? Nobel in AI Project

Unit 4: The Fabric of the Internet and AI

wk	
1	Introduction to the Internet Sending Bits IP Addresses Routers and Redundancy Packets
2	HTTP and DNS The Digital Divide Internet Access Advocacy Campaign
3	Network Security Emerging Technologies Network Security Risk Assessment

Unit 5: Cybersecurity and Global Impacts

wk	
1	Innovation Simulation Part 1 Data Policies and Privacy The Value of Privacy Innovation Simulation Part 2
2	Security Risks Part 1 Security Risks Part 2 Innovation Simulation Part 3 Protecting Data Part 1 Protecting Data Part 2
3	Innovation Simulation Part 4

Unit 6: Insights from Data and AI

wk	
1	Introduction to Data Science Making Sense of Data Ethical Data Collection Data Rights Effective Data Questions
2	Data Storytelling Introduction to Data Visualization Data Cleaning and Analysis Techniques Data Interpretation More Visualization Techniques
3	Crafting Stories with Data

Code.org Computing Foundations for a Digital Age: Indiana (AI Foundations) Syllabus and Overview

Curriculum Outline

CFDA: Indiana (AIF) consists of six units designed to provide students with both foundational knowledge and hands-on experience. Each unit focuses on core computer science and AI topics, blending computational thinking with real-world applications. The following outlines the content of each unit.

Unit 1: Problem Solving with AI

15 (45 minute) class periods

Students learn how generative AI models produce text and images by identifying patterns and making predictions using training data and probability. They develop skills in strategic prompting by experimenting with input structure, context, and media type, and analyzing how these choices affect AI-generated outputs. Through comparisons of human and AI responses, students build awareness of AI's limitations and biases, and gain practice refining AI outputs to make them clearer, more accurate, and more aligned with user intent.

Week 1	**Lesson 1** Talking to Machines	**Lesson 2** Beyond Words	**Lesson 3** The AI's Brain	**Lesson 4** Smart or Just Predictable?	**Lesson 5** Uncovering Contradictions
Week 2	**Lesson 6** Understanding Bias	**Lesson 7** AI's Wild Imagination	**Lesson 8** Debugging and Refining Outputs	**Lesson 9** AI as a Co-Creator	**Lesson 10** Remixing, Creativity, and Originality
Week 3	**Lesson 11** Ownership, Ethics, and Creativity	**Lesson 12** AI's Role in Society	**Lesson 13** AI Time Capsule Project		

Learning Outcomes

By the end of this unit, students will be able to . . .

- Explain how AI generates responses using training data, probability, and neural networks.
- Analyze AI-generated content to identify bias, hallucinations, and misinformation.
- Apply debugging techniques to refine AI outputs and improve response accuracy.
- Use AI as a collaborative tool while recognizing its limitations and risks.
- Evaluate the ethical implications of AI in decision-making and creative fields.

Unit Project

In this project, students will curate historical data and use AI to generate future predictions in areas like technology, fashion, media, healthcare, and climate policy. They will analyze AI's assumptions, biases, and

ethical considerations, refining AI-generated outputs and proposing responsible AI applications. The project culminates in a presentation or multimedia artifact showcasing both AI-generated content and students' critical evaluation of AI's reasoning, limitations, and biases.

Indiana Computing Foundations for a Digital Age Standards

- 4565.D1.9 Assess a program by testing to verify correct behavior.
- 4565.D2.6 Analyze data using computational thinking principles to make inferences or predictions.
- 4565.D2.8 Assess whether and how a given question can be answered using computational methods and data, and what specific data is needed.
- 4565.D4.7 Use documentation and other resources to guide tasks such as installation and troubleshooting.
- 4565.D5.2 Discuss the laws surrounding intellectual property.
- 4565.D5.4 Examine how emerging technologies are impacting a variety of practices (e.g., use of facial recognition in policing, AI-generated news products).
- 4565.D5.5 Evaluate the use of emerging technologies (e.g., generative AI) for accuracy and to meet specific needs.

CSTA Standards

- 3A-CS-01 Explain how abstractions hide the underlying implementation details of computing systems.
- 3A-IC-24 Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices.
- 3A-IC-25 Test and refine computational artifacts to reduce bias and improve fairness.
- 3A-AP-22 Design and develop computational artifacts using collaborative tools.
- 3A-AP-23 Document design decisions using text, graphics, presentations, and/or demonstrations.

Unit 2: Foundations of AI Programming

15 (45 minute) class periods

Students develop foundational programming skills in Python and practice computational thinking through creative, visual problem solving. Using the **`Painter`**, they learn how to sequence instructions, define and call functions, use loops to reduce repetition, and apply conditionals to make decisions. By solving challenges in The Neighborhood, students build confidence as programmers while exploring how AI systems follow rules and logic to complete tasks. Throughout the unit, students reflect on their own problem-solving strategies, consider the role of computing in different careers, and begin to see themselves as creators of technology.

Week 1	**Lesson 1** Computing Careers	**Lesson 2** Computational Thinking	**Lesson 3** Designing Algorithms	**Lesson 4** Variables and Data Flow	**Lesson 5** Working with Objects
Week 2	**Lesson 6** Functions	**Lesson 7** Functions with Parameters	**Lesson 8** Introduction to Loops	**Lesson 9** Conditional Statements	**Lesson 10** Algorithms IRL
Week 3	**Lesson 11** Decomposition	**Lesson 12** Two-Way Selection	**Lesson 13** Pixels of Me Project		

Learning Outcomes

By the end of this unit, students will be able to . . .

- Write and sequence Python commands.
- Define and call functions to break problems into reusable parts.
- Use loops and conditionals to make programs more efficient and responsive.
- Apply computational thinking strategies like decomposition and pattern recognition to solve programming challenges.
- Reflect on how AI systems follow instructions and make decisions using rules and logic.

Unit Project

In this project, students design a digital artifact using Python that represents an aspect of their identity, such as a hobby, career aspiration, or favorite story. They will plan their design with pseudocode, break it into components, and implement it using functions, loops, and conditionals. The project emphasizes creativity, collaboration, and technical skill development while integrating ethical reflections on AI's role in programming, encouraging students to apply computational thinking to create meaningful and personalized designs.

Code.org Computing Foundations for a Digital Age: Indiana (AI Foundations) Syllabus and Overview

Indiana Computing Foundations for a Digital Age Standards

- 4565.D1.1 Define algorithm and explain what algorithms are used for.
- 4565.D1.2 Describe the difference between traditional algorithms and artificial intelligence/machine learning (AI/ML) algorithms and, at a high level, describe how AI/ML algorithms work.
- 4565.D1.3 Explain why/how sequence matters in an algorithm.
- 4565.D1.4 Interpret and modify algorithms (e.g., to add functionality).
- 4565.D1.5 Compare (at a high level) the trade-offs (e.g., speed, memory) of different algorithms.
- 4565.D1.6 Reference documentation and other online tools to assist with programming.
- 4565.D1.7 Interpret the function of a segment of code and convert an algorithm to code.
- 4565.D1.8 Formulate algorithms using programming structures to decompose a complex problem.
- 4565.D1.10 Illustrate knowledge of good programming practice including the use of conventional standards and comments.
- 4565.D2.1 Identify and define data types (e.g., string, numeric, Boolean) and how it is created, stored, and used by computers.
- 4565.D5.4 Examine how emerging technologies are impacting a variety of practices (e.g., use of facial recognition in policing, AI-generated news products).

CSTA Standards

- 3A-AP-13 Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.
- 3A-AP-15 Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance, and explain the benefits and drawbacks of choices made.
- 3A-AP-16 Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.
- 3A-AP-17 Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.
- 3A-AP-19 Systematically design and develop programs for broad audiences by incorporating feedback from users.
- 3A-AP-22 Design and develop computational artifacts working in team roles using collaborative tools.
- 3A-AP- 23 Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.
- 3A-IC-24 Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices.
- 3A-IC-25 Test and refine computational artifacts to reduce bias and equity deficits.
- 3A-IC-26 Demonstrate ways a given algorithm applies to problems across disciplines.

Unit 3: AI and the Systems That Power It

15 (45 minute) class periods

Students investigate how computing systems work by exploring the roles of hardware and software in processing information. They analyze how inputs, outputs, storage, and processing interact within real-world systems and AI-powered technologies. Through hands-on exploration and debugging challenges, students build mental models of how computers and devices function, practice abstraction to manage complexity, and reflect on how their personal experiences and values can shape future pathways in computing.

Week 1	**Lesson 1** Introduction to Computer Systems	**Lesson 2** Hardware	**Lesson 3** Troubleshooting and Optimizing Hardware	**Lesson 4** Software and Operating Systems	**Lesson 5** Troubleshooting and Optimizing Software
Week 2	**Lesson 6** Build a Computer System	**Lesson 7** Data in Computer Systems	**Lesson 8** Processing Data in a System	**Lesson 9** How AI Uses Data	**Lesson 10** IoT and Emerging Technologies
Week 3	**Lesson 11** User Testing	**Lesson 12** Just Because We Can, Should We?	**Lesson 13** Nobel in AI Project		

Learning Outcomes

By the end of this unit, students will be able to . . .

- Describe how hardware and software components work together to input, store, process, and output information.
- Identify examples of computing systems in everyday life and explain how they support AI applications.
- Use abstraction and decomposition to explore how systems function and how problems are solved.
- Apply debugging strategies to identify and fix issues in systems.
- Explore computing careers and reflect on how personal interests and values connect to different pathways in tech.

Unit Project

In this project, students design an innovative computer system to address one of the UN 's Sustainable Development Goals. They will create a product that explores and integrates user profiles, relevant data, computer system flowcharting, and the ethical considerations surrounding their solution. The product will explain how the system operates across computational layers (hardware, software, network, etc.) and

addresses the needs of the target users. Throughout the process, students will engage with AI in a collaborative fashion in order to provide depth, accuracy, and creativity to their proposed innovation.

Indiana Computing Foundations for a Digital Age Standards

- 4565.D1.6 Reference documentation and other online tools to assist with programming.
- 4565.D2.1 Identify and define data types (e.g., string, numeric, Boolean) and how it is created, stored, and used by computers.
- 4565.D2.4 Describe how different types of data (e.g., audio, visual, spatial, environmental) can be collected computationally.
- 4565.D2.6 Analyze data using computational thinking principles to make inferences or predictions.
- 4565.D2.8 Assess whether and how a given question can be answered using computational methods and data, and what specific data is needed.
- 4565.D3.1 Demonstrate awareness of the history of computing.
- 4565.D3.5 Discuss the ethical and appropriate use of computer devices and examine device usability through several lenses including accessibility, ergonomics, and learnability.
- 4565.D4.2 Identify various types of hardware (including components) and software (including operating systems) and explore the security practices, functionality, cost, accessibility, and aesthetics of a variety of hardware and software
- 4565.D4.4 Explain how an operating system, other software, and hardware work together.
- 4565.D4.5 Describe why cybersecurity is important and evaluate the social and emotional implications of privacy in the context of safety, law, and ethics.
- 4565.D4.6 Optimize operating systems and other software settings to achieve goals.
- 4565.D4.7 Use documentation and other resources to guide tasks such as installation and troubleshooting.
- 4565.D5.1 Explain the privacy concerns related to the collection and generation of data through implicit and explicit processes.
- 4565.D5.4 Examine how emerging technologies are impacting a variety of practices (e.g., use of facial recognition in policing, AI-generated news products).
- 4565.D5.5 Evaluate the use of emerging technologies (e.g., generative AI) for accuracy and to meet specific needs.

CSTA Standards

- 3A-CS-01 Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects.
- 3A-CS-02 Compare levels of abstraction and interactions between application software, system software, and hardware layers.
- 3A-CS-03 Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.
- 3A-DA-10 Evaluate the tradeoffs in how data elements are organized and where data is stored.

- 3A-AP-13 Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.
- 3A-AP-22 Design and develop computational artifacts working in team roles using collaborative tools.
- 3A-AP-23 Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.
- 3A-IC-24 Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices.
- 3A-IC-25 Test and refine computational artifacts to reduce bias and equity deficits.

Unit 4: The Fabric of the Internet and AI

15 (45 minute) class periods

Students explore how the Internet works by examining how data travels across physical and logical networks using protocols like TCP/IP, HTTP, and DNS. They analyze how addressing systems, packets, and redundancy enable reliable communication, and investigate how AI improves network performance, accessibility, and information retrieval. Through simulations and real-world examples, students develop a systems-level understanding of digital communication and consider the societal impact of equitable Internet access and the role of intelligent systems in shaping their online experiences.

Week 1	**Lesson 1** Introduction to the Internet	**Lesson 2** Sending Bits	**Lesson 3** IP Addresses	**Lesson 4** Routers and Redundancy	**Lesson 5** Packets
Week 2	**Lesson 6** HTTP and DNS	**Lesson 7** The Digital Divide	**Lesson 8** Internet Access Advocacy Campaign Project		
Week 3	**Lesson 9** Network Security	**Lesson 10** Emerging Technologies	**Lesson 11** Network Security Risk Assessment Project		

Learning Outcomes

By the end of this unit, students will be able to . . .

- Explain how data is transmitted across networks using addressing, packets, and common protocols.
- Describe how the Internet's layered structure supports reliable and scalable communication.
- Identify how AI is used to improve Internet systems, including content filtering, routing, and accessibility.
- Analyze how design decisions in Internet infrastructure affect access, speed, and reliability.
- Evaluate the societal impact of networked technologies and the role of AI in shaping online experiences.

Unit Projects

This unit offers students two projects that provide hands-on opportunities to apply their knowledge of networks and the Internet to real-world challenges:

1. Internet Access Advocacy Campaign: In this project, students act as advocacy groups or policy advisors, tasked with creating a multimedia campaign to promote equitable Internet access. The campaign addresses

key issues such as the digital divide, net neutrality, and privacy concerns, while highlighting the role of emerging technologies, like AI, in enhancing Internet infrastructure. Students develop persuasive content tailored for an audience of policymakers or community leaders, proposing actionable solutions to bridge Internet access gaps. This project emphasizes creativity, ethical thinking, and the importance of inclusive technology policies, preparing students to understand the societal impacts of network decisions.

2. Network Security Risk Assessment: In this project, students take on the role of security consultants tasked with evaluating the vulnerabilities of a company's network. They analyze the network's infrastructure, identifying potential weaknesses, such as encryption gaps or unsecured devices, while considering the societal and ethical implications of data breaches. Students propose AI-driven solutions for real-time threat detection, monitoring, and response to enhance network security. The project culminates in a professional risk assessment report and presentation to a simulated leadership team, teaching students how to communicate technical recommendations effectively. This project emphasizes problem-solving, technical analysis, and responsible network management.

Indiana Computing Foundations for a Digital Age Standards

- 4565.D1.2 Describe the difference between traditional algorithms and artificial intelligence/machine learning (AI/ML) algorithms and, at a high level, describe how AI/ML algorithms work.
- 4565.D3.2 Evaluate the scalability and reliability of networks, by describing the relationship between routers, switches, servers, topology, and addressing.
- 4565.D3.3 Compare various security measures, considering tradeoffs between the usability and security of a computing system.
- 4565.D3.4 Explain tradeoffs when selecting and implementing cybersecurity recommendations.
- 4565.D3.5 Discuss the ethical and appropriate use of computer devices and examine device usability through several lenses including accessibility, ergonomics, and learnability.
- 4565.D3.6 Examine the impact of the Internet on society.
- 4565.D4.3 Explain what networks (including the Internet) are and explore the fundamental principles and components of computer networking.
- 4565.D4.5 Describe why cybersecurity is important and evaluate the social and emotional implications of privacy in the context of safety, law, and ethics.
- 4565.D5.5 Evaluate the use of emerging technologies (e.g., generative AI) for accuracy and to meet specific needs.

CSTA Standards

- 3A-NI-04 Evaluate the scalability and reliability of networks, by describing the relationship between routers, switches, servers, topology, and addressing.
- 3A-NI-05 Give examples to illustrate how sensitive data can be affected by malware and other attacks.
- 3A-AP-22 Design and develop computational artifacts working in team roles using collaborative tools.
- 3A-IC-24 Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices.

Unit 5: Cybersecurity and Global Impacts

13 (45 minute) class periods

This unit introduces students to cybersecurity and the global impact of computing innovations through a "Future School" simulation project, where they research and propose technological enhancements for school life. Students examine data collection, privacy policies, and the trade-offs between convenience, security, and privacy. Hands-on activities, including public service announcements and security risk investigations, help them understand threats like phishing, keylogging, and malware, along with protective measures such as encryption and multifactor authentication. Grounded in ethical computing and computational thinking, this unit fosters analytical skills, teamwork, and ethical awareness, preparing students to navigate and contribute responsibly to the evolving digital world.

Learning Outcomes

By the end of this unit, students will be able to . . .

- Identify and explain the benefits of computing innovations in different contexts.
- Evaluate the unintended consequences that may arise from the use of computing innovations.
- Define common privacy and security risks, such as phishing, keylogging, and malware.
- Describe how personal information can be exploited and the potential impacts on individuals.
- Analyze scenarios to weigh the benefits of convenience against potential risks to privacy and security.
- Develop informed opinions on privacy trade-offs and justify personal perspectives.
- Collaborate to design computing innovations that address specific problems.
- Defend proposals by considering their ethical implications, benefits, and potential risks.

Unit Project

The Innovation Simulation project serves as the culminating experience for this unit, immersing students in a real-world scenario where they act as stakeholders at a "Future School" convention. In this simulation,

students work in teams to design and propose computing innovations that enhance school life while addressing potential security, privacy, and ethical challenges. This project allows students to apply the concepts they've learned throughout the unit—such as encryption, data privacy, and network security—in a practical, collaborative setting.

Indiana Computing Foundations for a Digital Age Standards

- 4565.D3.1 Demonstrate awareness of the history of computing.
- 4565.D3.2 Evaluate the scalability and reliability of networks, by describing the relationship between routers, switches, servers, topology, and addressing.
- 4565.D3.3 Compare various security measures, considering tradeoffs between the usability and security of a computing system.
- 4565.D3.4 Explain tradeoffs when selecting and implementing cybersecurity recommendations.
- 4565.D4.1 Examine the dynamic between privacy and security.
- 4565.D4.5 Describe why cybersecurity is important and evaluate the social and emotional implications of privacy in the context of safety, law, and ethics.
- 4565.D5.1 Explain the privacy concerns related to the collection and generation of data through implicit and explicit processes.
- 4565.D5.3 Examine tradeoffs in computing technologies through current events related to broad ideas including privacy, communication, and automation (i.e., driverless cars can increase convenience and reduce accidents, but they are susceptible to hacking. The emerging industry will reduce the number of taxi and ride-share drivers but will create software engineering and cybersecurity jobs).

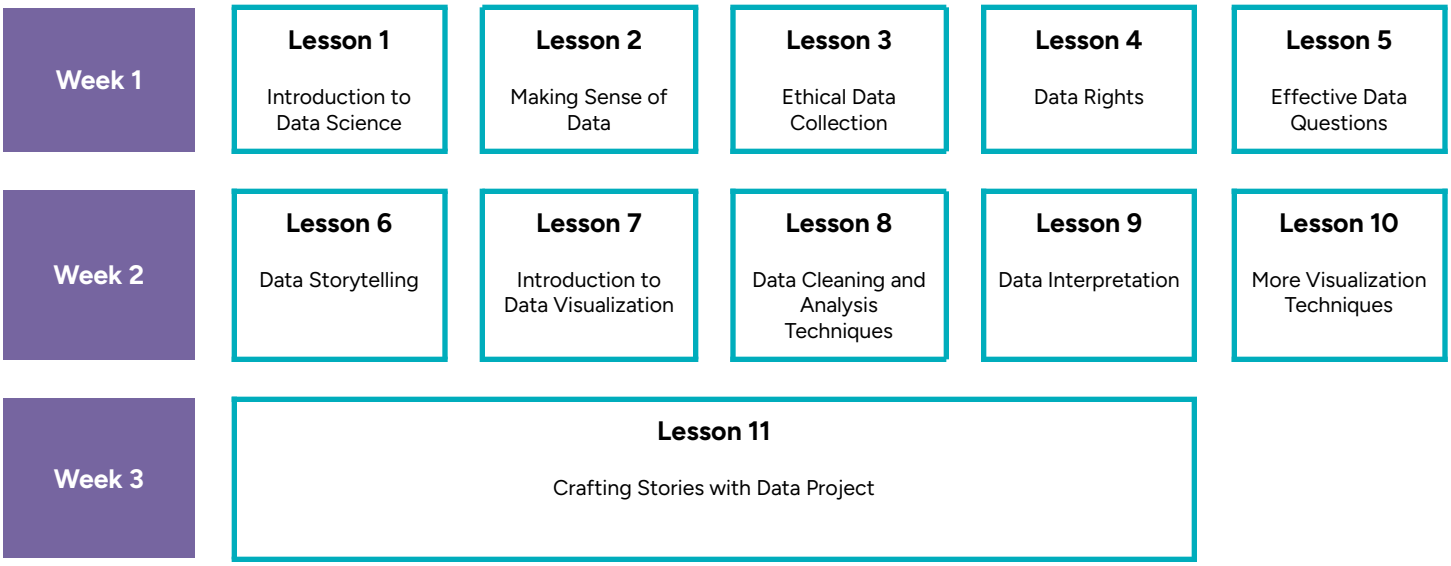
CSTA Standards

- 3A-NI-06 Recommend security measures to address various scenarios based on factors such as efficiency, feasibility, and ethical impacts.
- 3A-NI-07 Compare various security measures, considering tradeoffs between the usability and security of a computing system.
- 3A-NI-08 Explain tradeoffs when selecting and implementing cybersecurity recommendations.
- 3A-AP-22 Design and develop computational artifacts working in team roles using collaborative tools.
- 3A-IC-24 Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices.

Unit 6: Introduction to Data Science

14 (45 minute) class periods

Students build foundational data science skills by collecting, organizing, visualizing, and interpreting data to answer meaningful questions. They learn how to clean and analyze datasets using tools like charts and summaries, and reflect on the role of data in shaping understanding. Throughout the unit, students explore how AI systems analyze data at scale, make predictions, and influence decision-making. By engaging with real-world examples, students practice ethical reasoning, evaluate the limitations of data and algorithms, and begin to see data as a tool for inquiry, insight, and impact.



Learning Outcomes

- Ask and refine questions that can be answered with data.
- Collect, clean, and organize data to prepare it for analysis.
- Use visualizations to identify patterns and communicate insights.
- Interpret data in context and evaluate the strengths and limitations of conclusions.
- Explain how AI systems use data to make predictions and support decision-making.

Unit Project

The culminating project in this unit challenges students to explore a personally relevant or community-based issue through data analysis, transforming raw data into meaningful narratives. This project emphasizes the importance of data literacy by guiding students through the data science process—from identifying research questions to visualizing trends and presenting insights. Students gain hands-on experience working with real datasets while developing storytelling skills that connect data to real-world issues, fostering critical thinking and communication.

Code.org Computing Foundations for a Digital Age: Indiana (AI Foundations) Syllabus and Overview

Indiana Computing Foundations for a Digital Age Standards

- 4565.D2.2 Identify basic data formats (e.g., tables, schemas, JSON) and how computers represent data.
- 4565.D2.3 Understand the difference between data and metadata.
- 4565.D2.4 Describe how different types of data (e.g., audio, visual, spatial, environmental) can be collected computationally.
- 4565.D2.5 Transform and prepare (e.g., normalize, merge, clean) data visualizations, models, and simulations using data collected using computational tools such as surveys.
- 4565.D2.6 Analyze data using computational thinking principles to make inferences or predictions.
- 4565.D2.7 Evaluate approaches to cleaning data in a given context.
- 4565.D2.8 Assess whether and how a given question can be answered using computational methods and data, and what specific data is needed.
- 4565.D4.1 Examine the dynamic between privacy and security.
- 4565.D4.5 Describe why cybersecurity is important and evaluate the social and emotional implications of privacy in the context of safety, law, and ethics.
- 4565.D5.1 Explain the privacy concerns related to the collection and generation of data through implicit and explicit processes.
- 4565.D5.4 Examine how emerging technologies are impacting a variety of practices (e.g., use of facial recognition in policing, AI-generated news products).
- 4565.D5.5 Evaluate the use of emerging technologies (e.g., generative AI) for accuracy and to meet specific needs.

CSTA Standards

- 3A-DA-11 Create interactive data visualizations using software tools to help others better understand real-world phenomena.
- 3A-AP-13 Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.
- 3A-AP-22 Design and develop computational artifacts working in team roles using collaborative tools.
- 3A-AP-23 Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.
- 3A-IC-24 Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices.

Semester 2 (Coming Soon!)

This semester offers an innovative approach to teaching programming by integrating learning with and about artificial intelligence (AI). Using Python as the primary language, students build foundational programming skills while leveraging AI tools to enhance computational thinking and problem-solving. The curriculum also introduces students to the basics of creating AI-powered programs, exploring machine learning, and applying data science principles.

Key Features

- **AI-Enhanced Programming:** Students use AI tools to support coding, debugging, and algorithm design, enabling them to think critically about computational processes and outcomes.
- **Foundational Skills in Python:** Students will write, test, and refine Python programs, applying constructs like variables, loops, conditionals, and functions to solve real-world problems.
- **Learning About AI:** The curriculum introduces students to fundamental AI concepts, including machine learning and data analysis, through hands-on programs that demonstrate how AI systems are designed and applied.
- **Real-World Relevance:** Students develop projects that address meaningful challenges, integrating programming and AI to create impactful solutions while reflecting on ethical and societal implications.
- **Ethical and Computational Thinking:** Through guided discussions and activities, students analyze issues such as bias, privacy, and fairness in AI, preparing them to engage responsibly with emerging technologies.
- **Career Exploration:** The course highlights the diverse applications of programming and AI across industries, inspiring students to envision pathways in technology-related careers.

By the end of Semester 2, students will not only gain proficiency in programming but also develop a foundational understanding of AI systems and their role in shaping the future.