

# Arkansas Introduction to Computer Science Course Syllabus

## Course Overview

This course introduces students to the foundations of computer science while highlighting its relevance across industries and career pathways. Over the span of 25–30 weeks, students will explore computational thinking, programming in Python, data analysis, and the ethical and societal impacts of technology. They will build professional skills through portfolio development, practice abstraction and problem-solving, and learn to design, debug, and document code. Students will also work with data files and visualization tools, evaluate emerging technologies such as AI and cloud computing, and examine principles of cybersecurity, systems, and networking. By the end of the course, students will have created a portfolio of authentic projects that demonstrate both technical proficiency and professional readiness, preparing them for further study and careers in computer science.

## Prerequisites

This course is designed for students with no prior computer science or programming experience. However, a strong foundation in high school math and problem-solving skills are recommended to support success.

## Required Resources

- Online IDE & Curriculum: [JuiceMind.com](https://juicemind.com)
- Python Programming Environment: Access to JuiceMind’s browser-based IDE (no local installation required)
- Supplemental Materials: [Arkansas CS Standards reference](#)

## Course Goals

- Introduce students to computer science careers and professional skills.
- Build foundational programming knowledge using Python.
- Develop computational thinking through abstraction, decomposition, and algorithm design.
- Strengthen problem-solving by applying coding to authentic, real-world projects.
- Analyze, interpret, and communicate insights from data using modern tools.
- Explore major and emerging technologies, including AI, cloud computing, IoT, and cybersecurity.
- Foster teamwork, communication, and professional readiness through collaborative projects.
- Promote ethical, secure, and responsible use of computing systems and innovations

## Learning Environment

The course is structured as an interactive, project-based learning experience that blends digital tools with in-class collaboration. Students will use JuiceMind's web-based IDE to complete coding exercises, build personal portfolio sites, and develop authentic projects. Lessons include hands-on programming in Python, unplugged activities to strengthen computational thinking, data analysis and visualization tasks, and explorations of emerging technologies. Students will also engage in written reflections, digital presentations, and group projects that emphasize professional communication, ethical practice, and teamwork.

## **Programming Environment**

Students will use JuiceMind’s online IDE to write, test, and debug Python programs. This browser-based environment makes it easy to collaborate, share code, and build projects while developing computational thinking skills.

## **Pacing**

Lesson plans are structured around a 45-minute class period, and each lesson folder is designed to be completed within a single period unless otherwise stated. The instructional content of a lesson will not exceed 45 minutes, though exercises, projects, and review may extend into homework depending on student ability.

In total, Introduction to Computer Science – Arkansas (2025 Revision) is designed to take approximately 100 class periods (25–30 weeks) to complete using only the materials included within this course. This is shorter than a typical school year to allow for additional review, enrichment, and alignment with state testing schedules.

## **Course Breakdown**

### **Unit 1: Careers & Professionalism (10 periods – 7 hrs 30 mins)**

Students are introduced to computer science as a career field and develop professional readiness skills. They will explore career pathways, practice professional communication, examine ethics in computing, and begin building an online portfolio.

- Topics Covered:
  - CS Career Pathways
  - Professional Communication
  - Ethics in Technology
  - Building an Online Portfolio
- Featured Activity: 1.4.3: Your Digital Portfolio Challenge [Students design a professional portfolio strategy to showcase their strengths, interests, and commitment to growth]

## **Unit 2: Computational Thinking & Problem-Solving** (10 periods – 7 hrs 30 mins)

Students develop computational thinking practices by applying abstraction, decomposition, and algorithm design to real-world problems.

- Topics Covered:
  - Computational Thinking Pillars
  - Pseudocode & Flowcharts
  - Pattern Recognition & Decomposition
- Featured Activity: 2.2.3: Activity Recommendation Algorithm [Students apply storyboards, flowcharts, and pseudocode to analyze and improve an algorithm for choosing activities based on preferences and conditions.]

## **Unit 3: Programming Fundamentals with Python** (12 periods – 9 hrs)

Students begin coding in Python by learning syntax, variables, data types, and operators. They will practice debugging and documenting their code.

- Topics Covered:
  - Code & Console Basics
  - Variables & Data Types
  - Mathematical & String Operators
  - Input & Output
- Sample Activity: 3.5.3: Exercise - Basketballs and Footballs [Students will write a program to track basketballs and footballs using variables, update the values, and print the results before and after the change.]

#### **Unit 4: Control Structures in Python** (10 periods – 7 hrs 30 mins)

Students learn how to implement decision-making and iteration in Python programs.

- Topics Covered:
  - Boolean Logic
  - Logical and Comparison Operators
  - Conditional Statements: If / If-Else
  - Specific Ranges
  - For & While Loops
- Featured Activity: 4.6.3: Exercise - Text Generator [Students will write a program that repeats a word provided by the user.]

#### **Unit 5: Functions & Data Structures** (10 periods – 7 hrs 30 mins)

Students create reusable functions and work with lists and dictionaries to manage collections of data.

- Topics Covered:
  - Defining Functions
  - Debugging Strategies
  - Lists & Dictionaries
- Featured Activity: 5.2.3: Exercise - Buggy Processing [Students will rewrite code to divide any valid pair of numbers while handling invalid inputs.]

**Midterm 1** (2 periods – 1 hr 30 mins) \*To be added at a later date

Covers Units 1–5 through multiple-choice and live-coding tasks.

**Unit 6: Working with Data & Files** (8 periods – 6 hrs)

Students learn how to read, write, and manipulate files in Python.

- Topics Covered:
  - Console I/O Revisited
  - File Types & Formats
  - File Operations in Python
- Sample Activity: 6.1.3: Console Chat Bot Challenge [Students will create a simple chatbot that uses input, print, and data conversion to hold a friendly conversation with users.]

**Unit 7: Data Analysis & Visualization** (15 periods – 11 hrs 15 mins)

Students collect, clean, and analyze data using Python, then present findings with visualizations and ethical reflection.

- Topics Covered:
  - Data Collection & Cleaning
  - Descriptive Statistics
  - Visualizations & Dashboards
  - Communicating Insights & Bias
- Featured Activity: 7.2.3: Cleaning the School Database [Students will identify and fix data quality issues in messy student survey data to ensure accurate analysis and actionable insights for improving school programs.]

### **Unit 8: Major & Emerging Technologies (8 periods – 6 hrs)**

Students explore AI, cloud computing, IoT, big data, and other emerging technologies, considering their societal impacts.

- Topics Covered:
  - Artificial Intelligence
  - Cloud Computing
  - IoT & Big Data
  - Emerging Technologies
- Sample Activity: 8.4.3: Future Tech Challenge [Students will act as tech consultants by identifying a community problem and proposing a solution using quantum computing, edge computing, XR, or blockchain.]

**Midterm 2 (2 periods – 1 hr 30 mins) \*To be added at a later date**

Covers Units 6–8 with scenario-based questions and coding tasks.

## **Unit 9: Cybersecurity, Systems & Networking (12 periods – 9 hrs)**

Students learn the fundamentals of cybersecurity, system hardware/software, and networking.

- Topics Covered:
  - Cybersecurity Principles & Threats
  - Hardware & Software Components
  - Networking & Security
- Featured Activity: 9.1.3: Digital Security Detective [Students will analyze security scenarios and apply cybersecurity fundamentals to identify threats and recommend protections for their school's digital environment.]

**Final Exam** (2 periods – 1 hr 30 mins) \*To be added at a later date

Comprehensive final exam covering the full course with both multiple-choice and coding sections.