Maryland Computer Programming 1 Syllabus

Course Overview

This course introduces students to foundational concepts in computer science through problem-solving, programming in Python, data exploration, and the societal impacts of computing. Students will design, implement, test, and explain programs while developing computational thinking, ethical awareness, and collaboration skills aligned to Maryland Computer Science standards.

Pre-requisites

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
- Python Programming Environment: Access to JuiceMind's browser-based IDE (no local installation required)
- Supplemental Materials: Maryland 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.

Course Goals (continued)

- 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.

Course Breakdown

Unit 1: Careers and Professionalisms

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
    - Building an Online Portfolio
    - Build a resume

Unit 2: The Internet and the impacts of Computing

Students explore how information is represented, transmitted, and used in modern computing systems. They will examine how the internet works, consider ethical and societal impacts of technology, and analyze how computing innovations influence communities, careers, and everyday life.

- Topics Covered:
    - Digital Information
    - Welcome to the internet
    - Ethics in Computing
    - Computing innovations

Unit 3: Solving Problems in Computer Science

Students are introduced to the basics of computational problem-solving, including algorithmic thinking and flowchart design.

- Topics Covered:
    - Problem-Solving Strategies
    - Algorithms

Unit 4: Programming fundamentals

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
- Featured activity: Mad Libs: Students have the opportunity to work in groups to design and implement the Mad-libs game in Python.

Unit 5: Control Structures

Students learn to implement control structures in Python, including conditional logic and loops.

- Topics Covered:
    - Boolean Logic
    - Conditional Statements
    - For Loops and While Loops
- Featured activity : Guess your classmate: Students have the opportunity to work in groups in a collaborative and inclusive development platform in order to design a game that prompts questions, to see if the player can guess who the correct classmate the game is referring to.

Unit 6: Libraries and Functions

Students learn how to organize and simplify programs by using functions and libraries. They will break complex problems into smaller tasks, reuse code efficiently, and apply built-in Python modules to create more powerful, readable, and maintainable programs while practicing debugging and testing strategies.

- Topics Covered:
    - Defining and Calling Functions
    - Parameters and Arguments
    - Return Values
    - Randomness and Libraries
    - Bugs

Unit 7: Data Structures

Students explore Python's data structures, such as lists, tuples, and strings, and learn how to manipulate these structures to solve problems.

- Topics Covered:
    - Lists
    - Tuples and Strings
    - Iterating through lists
    - List methods
- Featured activity: Grade calculator: [Algorithms and Programming (AAP), Computational Thinking Practice 3]: Students have the opportunity to build a grade calculator program that calculates various statistics and assigns letter grades based on an input. Students will implement abstractions in order to develop this program.

Unit 8: Working with Data & Files

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

- Topics Covered:
    - Console I/O revisited
    - Understanding file types
    - File operations in Python

Unit 9: Working with graphics in Python

Students explore how computer programs create visual and interactive experiences. By working with graphics, mouse events, and key events in Python, they will build programs that respond dynamically to user actions.

- Topics Covered:
    - Graphics in Python
    - Mouse Events
    - Key Events

Unit 10: Final Project: Build a game

Students will design and build an original interactive game using Python. They will apply programming concepts learned throughout the course—including variables, conditionals, loops, functions, data structures, and user input—to solve a real problem through game design. Students will plan, implement, test, and refine their game, and explain how their program works and why specific design decisions were made.

The final project emphasizes creativity, problem-solving, and computational thinking. Students will demonstrate their ability to organize code, handle user interaction, debug errors, and produce a complete, playable program. This project serves as preparation for performance-based assessments and showcases each student's Python programming skills.