Exploring Algorithms and Data with the AP Computer Science Principles (APCSP) Create Task

Unit Duration: 7 weeks

Unit Overview:
In this unit, students will delve into algorithms and data structures as they prepare to tackle the AP Computer Science Principles (APCSP) Create Task. They will explore various algorithms, data types, and programming concepts essential for understanding and implementing solutions to computational problems. Through hands-on activities, collaborative projects, and coding exercises, students will develop their computational thinking skills and gain proficiency in designing and implementing algorithms and data structures. Students will produce the APCSA Create Task and all of its components.

Unit Objectives:
- Understand fundamental algorithms and data structures.
- Analyze and evaluate algorithms for efficiency and effectiveness.
- Design and implement algorithms to solve computational problems.
- Utilize appropriate data structures to organize and manipulate data.
- Develop and refine computational solutions using iterative processes.
- Communicate and justify design decisions in creating computational artifacts.

Week 1: Introduction to Algorithms and Problem Solving
Introduce the concept of algorithms and their importance in problem-solving.
Discuss problem-solving strategies such as breaking problems into smaller subproblems, pattern recognition, and abstraction.
Explore examples of common algorithms (e.g., searching, sorting) and their applications.
Engage students in hands-on activities and coding exercises to practice algorithm design and implementation.
Assign readings and videos on algorithms and problem-solving techniques for further exploration.

Week 2: Data Structures and Computational Complexity
Introduce various data structures (lists, tuples, etc) and their characteristics.
Discuss the importance of selecting appropriate data structures based on problem requirements.
Create a practice Create Task with all elements from the rubric.
Engage students in coding exercises and problem-solving activities to apply concepts learned.
Assign projects where students must select and justify appropriate data structures for specific scenarios.

Week 3: Create Task Preparation
Review the requirements and rubric for the APCSP Create Task.
Create one minute videos to practice for the Create Task upload.
Provide guidance on selecting a project topic and designing a computational solution.
Facilitate brainstorming sessions and peer feedback for project ideas.
Review programs and answer the practice FRQ questions using given code (colab project).
Review programs and answer the practice FRQ questions using student written code.
Provide individualized support and guidance as students work on implementing their projects.
Conduct peer review sessions where students give and receive feedback on their project prototypes.
Encourage students to document their design decisions, algorithms, and data structures used in their projects.

**Week 4-7: Project Work**
Students are required to be given 9 hours of class time to complete the Create Task.
Allocated class time for students to work on their Create Task projects.

Assessment:
Formative assessments: In-class activities, coding exercises, and FRQ quizzes to assess understanding of concepts.
Summative assessments: Performance on the APCSP Create Task, including the completion and presentation of the computational artifact, documentation, and reflection (after the create task deadline)
Peer assessments: Peer review sessions for providing feedback on project prototypes and practice create tasks.