North Polk Community Schools Pacing & Course Description Guide

Grade Level:	Content:	Year:

Course Description/Rationale

AP Computer Science A is a two-semester course that introduces the student to procedure oriented programming. The student will be using Java as the programming language. This course will be the equivalent of the first semester of a college computer science curriculum. The course is recommended for students thinking of entering fields of computer science, mathematics, engineering, or any of the sciences. Students successfully completing this course may take the AP Computer Science A Exam to earn up to three (3) semester hours of college credit. Students will receive a weighted grade. (2 HS credits)

Name of Unit	Time Frame	Essential Learning Target	Standard(s)
Introduction to APCS	Week 1	Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.	3A-AP-13 Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests. (P5.2)
Using Objects in Java	Week 2-3	Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects.	3A-CS-01 Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects. (P4.1
Implementing	Week 3-5	Construct solutions to problems using student-created components, such as procedures, modules and/or objects.	3B-AP-14 Construct solutions to problems using student-created

			components, such as procedures, modules and/or objects.
Fundamental Data Types	Week 5-7	Use and adapt classic algorithms to solve computational problems.	3B-AP-10 Use and adapt classic algorithms to solve computational problems.
Decisions	Week 7-9	Control structures at this level may include conditional statements.	Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance, and explain the benefits and drawbacks of choices made.

Iteration	Week 9-11	Control structures at this level may include loops.	Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance, and explain the benefits and drawbacks of choices made.
Arrays & Array Lists	Week 11-13	Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.	Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.

Inheritance	Week 13-16	Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects using **inheritance.**	Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.
Interface & Polymorphism	Week 16	**Understand the reason for an interface in a large project.**	3B-AP-10 Use and adapt classic algorithms to solve computational problems.

Recursion	Week 17-19	Illustrate the flow of execution of a recursive algorithm.	3B-AP-13 Illustrate the flow of execution of a recursive algorithm.
Input/Output	Week 24	Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.	3A-AP-13 Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.
Object-AP Review	Week 20-36	**I can study for a major exam.**	3B-AP-15 Analyze a large-scale computational problem and identify generalizable patterns that can be applied to a solution.
Object-Oriented Design and Project	Week 24-36	Illustrate the flow of execution of a recursive algorithm.	3B-AP-17 Plan and develop programs for

			broad audiences using a software life cycle process.

