Pascack Valley Regional High School District

Pascack Hills High School, Montvale, New Jersey
Pascack Valley High School, Hillsdale, New Jersey

Course Name: AP Computer Science A

Born On: August, 2020
Previous Revision: August, 2022
Current Revision: August, 2023
Board Approval: 8/28/23

Course Description:

The purpose of this class is to introduce the student to the object-oriented programming paradigm using the Java language. This course emphasizes object-oriented methodology with a concentration on problem-solving and algorithm development. This course will follow the outline recommended by the AP® Computer Science Curriculum, Exam A. Concepts such as classes, objects, inheritance, polymorphism, and code reusability are studied. It also includes algorithms, data structures, and data abstraction. Hands-on laboratory work helps solidify each concept. In addition, students complete a long-term programming project that they must demonstrate in a formal presentation.

All computer science courses in the Pascack Valley Regional High School District are designed to address multiple learning styles and needs, and accommodations and modifications are made for students with disabilities, multilingual students, students at risk of failure, gifted and talented students, and students with 504 plans. AP Computer Science A builds on concepts learned and skills developed in Honors Computer Science, while also spiraling in those concepts and skills to reinforce and strengthen students' knowledge of computer science. Additionally, AP Computer Science A anticipates higher-level computer science concepts and skills that will be learned in Advanced Topics in Computer Science and college-level computer science courses, and enrichment opportunities are provided to challenge students and engage them in rich, interesting tasks. Students are encouraged to analyze data using tools and models to make valid and reliable claims (9.4.12.IML.3), and various technologies and applications are technology are integrated throughout the curriculum.

The Pascack Valley Regional High School Computer Science Department integrates 21st century life and career skills across its courses, with the dual goal of informing students about careers and fields of study that use computer science (9.3.ST.5, 9.3.ST-ET.5 and 9.3.ST-SM.2), and helping students improve the computational thinking skills they will need to in those careers and fields of study (9.2.12.CAP.2). Computer Science courses address the New Jersey Student Learning Standards for Career Readiness, Life Literacies and Key Skills, with a particular emphasis on demonstrating the ability to reflect, analyze and use creative skills and ideas (9.4.12.CI.1), investigating new challenges and opportunities for personal growth, advancement and transition (9.4.12.CI.3), identifying problem-solving strategies used in the development of an innovative product or practice (9.4.12.CT.1), and explaining the potential benefits of collaborating to enhance critical thinking and problem solving (9.4.12.CT.2). Computer Science courses also address the New Jersey Student Learning Standards for English Language Arts Companion Standards, with a particular focus on following complex multistep procedures (RST.9-10.3/RST.11-12.3), determining the meaning of symbols, key terms, and other domain-specific words and phrases (RST.9-10.4/RST.11-12.3), and translating quantitative or technical information expressed in words into visual forms and translating information expressed visually or mathematically into words (RST.9-10.7). Similarly, the Computer Science Department seeks to support students by providing them with opportunities to use computational thinking skills in interdisciplinary contexts, in contexts that are meaningful to students, and in contexts that attend to the contributions and perspectives of historically marginalized groups. Specifically, computer science courses will look to incorporate, when appropriate, contributions and experiences of people from the LGBTQ+ community and individuals with disabilities, and references to issues of social and cultural relevance, including climate change.

Course Outline [C2]

References to the following are abbreviated as follows:

(SG) -Study Guide **(TB)-** Text Book **(P)-**Programs

Curricular Requirements

[C2]- The course includes all of the topics listed in the “Computer Science A” column of the Topic Outline in the AP Computer Science Course Description.

[C3]- The course teaches students to design and implement computer-based solutions to problems in a variety of application areas.

[C4]—The course teaches students to use and implement commonly used algorithms and data structures.

[C5]—The course teaches students to develop and select appropriate algorithms and data structures to solve problems.

[C6]- The course teaches students to code fluently in an object-oriented paradigm using the programming language Java. The course teaches students to use standard Java library classes from the AP Java subset delineated in Appendices A and B of the *AP Computer Science Course Description*. (Note: Students who study a language other than Java in AP Computer Science must also be taught to use Java, as specified in the AP Java subset.)

[C7]—The course teaches students to read and understand a large program consisting of several classes and interacting objects, and enables students to read and understand the current *AP Computer Science Case Study* posted on AP Central®.

[C8]—The course teaches students to identify the major hardware and software components of a computer system, their relationship to one another, and the roles of these components within the system.

[C9]—The course teaches students to recognize the ethical and social implications of computer use.

AP Computer Science A: This course explores the object-oriented programming paradigm using the Java language, with a concentration on problem-solving and algorithm development. This course will follow the outline recommended by the AP® Computer Science Curriculum, Exam A. Concepts such as classes, objects, inheritance, polymorphism, and code reusability are studied. It also includes algorithms, data structures, and data abstraction.

Chapter in Horstmann Textbook/ Pacing Guide (Weeks)	**Key Learning Items/Concepts**	**NJSLS CS & Design Thinking Standards**	**Formative, Summative, Benchmark, and Alternative Assessments**	**Core Instructional and Supplemental Materials/ Modifications and Accommodations**
Unit 1 1 (1) Introduction 2 (2-3) Using Objects 3 (4-5) Implementing Classes 4 (6-7) Fundamental Data Types 5 (8-9) Decisions	**Week 1:** • Introduction to Computer Science and the Java Programming Language. Understand the activity of programming. • The anatomy of a computer. Learn about the architecture of Computers, including hardware and software components • Hardware/Software • Learn about machine code and high-level programming languages • Become familiar with Eclipse and the Computing environment • Errors • The Compilation Process: Compile and run programs • Recognize syntax and logic errors • Understand the file system and have a Backup Strategy **Weeks 2-3:** • Learn about the types and variables • The assignment operator • Classes. • Understand the concepts of classes and objects. • Be able to call methods. • Learn about parameters and return values. • Accessor and mutator methods • API documentation • Object References **Weeks 4-5:** • Black boxes • Public interfaces • Instance fields	**NJSLS Content Standards** 8.1.12.CS.1 8.1.12.CS.2 8.1.12.CS.3 8.1.12.CS.4 8.1.12.IC.1 8.1.12.DA.1 8.1.12.DA.2 8.1.12.DA.3 8.1.12.DA.4 8.1.12.AP.1 8.1.12.AP.2 8.1.12.AP.4 **NJSLS SMP** MP1. Make sense of problems and persevere in solving them MP2. Construct viable arguments and critique the reasoning of others MP3. Reason abstractly and quantitatively MP4. Model with mathematics MP5. Attend to precision MP6. Use appropriate tools strategically MP7. Look for and make use of structure MP8. Look for and express regularity in repeated reasoning	*Suggestion(s):* Students will be assessed regularly throughout this course, with a focus on both conceptual understanding and procedural fluency. Assessment tools may include the following: - quizzes (F) - tests (S) - performance tasks (F/S) - projects (S) - homework (F) - discussions (F) - journals (F) - Form A, B, or C benchmark (B) - alternative assessments (A) - portfolio (F, S) - online learning courses (F) - Group Learning Projects (F) - Individual Projects (S) - Oral Presentations (S)	Selection of primary sources *Suggestion(s):* College Board. *AP GridWorld Case Study.* New York: College Entrance Examination Board, 2006. (on grade level); Horstmann, Cay. *Java Concepts*. 4th ed. Hoboken, N.J.: Wiley, 2006. (on grade level; remediation); Trees, Frances P. *AP Computer Science Study Guide to Accompany Java Concepts*. 4th ed. Hoboken, N.J.: Wiley, 2006. (on grade level; remediation; advanced) **Programs:** P2.1-2.2 P2.9 P3.6, P3.7, P3.12 (Write small classes from scratch, choosing appropriate data representation.) Assign small programs that illustrate different types of input and output. Also P4.4-4.6 (DataSet) P4.12 (Digit Extractor) P6.4, P6.10,P6.14 **Modifications and Accommodations:** **Students with special needs:** Teachers and support staff will attend to all modifications and accommodations listed in students' IEPs and 504s. Teachers will incorporate

	• Implementing constructors and methods • Testing a class • Implicit and explicit method parameters • Understand the importance of comments **Weeks 6-7:** • Number Types • Recognize the importance of the numeric types • Assignment, increment, and decrement • Constants • Arithmetic and Mathematical Functions • Calling Static Methods • Strings • Review of Binary Numbers **Weeks 8-9:** • if statement • comparing values • Multiple Alternatives • Boolean expressions **Content-specific modifications and accommodations** - use multiple representations and technology to support conceptual understanding - provide students with skeletons of code and/or utilize flexible grouping **Interdisciplinary/additional connections** - Engage in programming assignments that have applications in science, mathematics, or business - Consider potential programming applications for climate change - Draw on contexts from diverse groups for programming exercises	1. Fostering an inclusive Computing Culture 2. Collaborating around Computing 3. Recognizing and Defining Computational Problems 7. Communicating about Computing **NJSLS for ELA Companion Standards** RST.9-10.3 RST.9-10.4 RST.9-10.7 RST.11-12.3 RST.11-12.4 **NJSLS-CLKS** **- 21st Century Life and Careers** 9.4.12.CI.1 9.4.12.CI.3 9.4.12.CT.1 9.4.12.CT.2 **- Technology** 9.4.12.IML.3 **- Career Education** 9.2.12.CAP.2 9.3.ST.5 9.3.ST-ET.5 9.3.ST-SM.2		manipulatives, extra time, alternative assessments, scaffolding, spiraling, technology, and flexible grouping to support student learning. **Multilingual students:** Teachers and support staff will work to support multilingual students in their first language and in English, providing materials and/or resources to support students' understanding. Students will be given additional time, as appropriate, and translation tools will be utilized as needed. **Students at risk of school failure:** Formative and summative data will be used to monitor student success, and students at risk of failure will receive additional supports and services, which may include parent consultation, extra help, and differentiation strategies, including small group instruction, group work, scaffolding, and spiraling. **Gifted and Talented Students:** Students who excel in their mastery of course standards will be further challenged with more complex tasks, extensions of concepts and skills, and extended problem solving and critical thinking opportunities.

AP Computer Science A: This course explores the object-oriented programming paradigm using the Java language, with a concentration on problem-solving and algorithm development. This course will follow the outline recommended by the AP® Computer Science Curriculum, Exam A. Concepts such as classes, objects, inheritance, polymorphism, and code reusability are studied. It also includes algorithms, data structures, and data abstraction.

Chapter in Horstmann Textbook/ Pacing Guide (Weeks)	**Key Learning Items/Concepts**	**NJSLS CS & Design Thinking Standards**	**Formative, Summative, Benchmark, and Alternative Assessments**	**Core Instructional and Supplemental Materials/ Modifications and Accommodations**
Unit 2 6 (10-12) Iteration 7 (13) Case Study Part 1- Consumer Review 8 (14-16) Arrays and ArrayLists 9 (17-19) Designing Classes	**Weeks 10-12:** • while loops • for loops • Nested loops • Avoid infinite loops and off-by- one errors • Processing sentinel values • Random Numbers and Simulation **Week 13:** • Experimenting with a large program • Using classes • Observe the attributes and the behavior of the actors **Weeks 14-16:** • Arrays and ArrayLists • Wrappers and Auto-Boxing • The generalized for loop • Study common Array Algorithms • Copying arrays • Understand when to choose the appropriate data structure in your programs. Arrays vs. array lists. **Weeks 17-19:** • Choosing appropriate classes • Cohesion and coupling • Accessors, mutators, and immutable classes • Side effects	**NJSLS Content Standards** 8.1.12.AP.1 8.1.12.AP.2 8.1.12.AP.3 8.1.12.AP.4 8.1.12.AP.5 8.1.12.AP.6 8.1.12.AP.7 8.1.12.AP.8 8.1.12.CS.4 8.1.12.DA.2 8.1.12.DA.6 **NJSLS SMP** MP1. Make sense of problems and persevere in solving them MP2. Construct viable arguments and critique the reasoning of others MP3. Reason abstractly and quantitatively MP4. Model with mathematics MP5. Attend to precision MP6. Use appropriate tools strategically MP7. Look for and make use of structure MP8. Look for and express regularity in repeated reasoning	*Suggestion(s):* Students will be assessed regularly throughout this course, with a focus on both conceptual understanding and procedural fluency. Assessment tools may include the following: - quizzes (F) - tests (S) - performance tasks (F/S) - projects (S) - homework (F) - discussions (F) - journals (F) - Form A, B, or C benchmark (B) - alternative assessments (A) - portfolio (F, S) - online learning courses (F) - Group Learning Projects (F) - Individual Projects (S) - Oral Presentations (S)	Selection of primary sources *Suggestion(s):* College Board. *AP GridWorld Case Study.* New York: College Entrance Examination Board, 2006. (on grade level); Horstmann, Cay. *Java Concepts*. 4th ed. Hoboken, N.J.: Wiley, 2006. (on grade level; remediation); Trees, Frances P. *AP Computer Science Study Guide to Accompany Java Concepts*. 4th ed. Hoboken, N.J.: Wiley, 2006. (on grade level; remediation; advanced) **Programs:** Roulette Program in SG, FunNumber Program, P7.15 Project 7.2 (NIM) P8.2-8.4,P8.10, Project 8.1 (Poker) (P) Concentration Create classes from scratch, Choosing appropriate data structures **Project 9.2** (Bar Codes/Zip Codes) **Modifications and Accommodations:** **Students with special needs**: Teachers and support staff will attend to all modifications and accommodations listed in students' IEPs and 504s. Teachers will incorporate manipulatives, extra time, alternative assessments, scaffolding, spiraling, technology, and flexible grouping to support student learning. **Multilingual students**: Teachers and support staff will work to support multilingual students in their first language and in English, providing materials and/or resources to support students' understanding. Students will be given additional time, as

	• Preconditions • Static methods • Static fields • Scope packages **Content-specific modifications and accommodations** - use multiple representations and technology to support conceptual understanding - provide students with skeletons of code and/or utilize flexible grouping **Interdisciplinary/additional connections** - Engage in programming assignments that have applications in science, mathematics, or business - Consider potential programming applications for climate change - Draw on contexts from diverse groups for programming exercises	1. Fostering an inclusive Computing Culture 2. Collaborating around Computing 3. Recognizing and Defining Computational Problems 7. Communicating about Computing **NJSLS for ELA Companion Standards** RST.9-10.3 RST.9-10.4 RST.9-10.7 RST.11-12.3 RST.11-12.4 **NJSLS-CLKS** **- 21st Century Life and Careers** 9.4.12.CI.1 9.4.12.CI.3 9.4.12.CT.1 9.4.12.CT.2 **- Technology** 9.4.12.IML.3 **- Career Education** 9.2.12.CAP.2 9.3.ST.5 9.3.ST-ET.5 9.3.ST-SM.2		appropriate, and translation tools will be utilized as needed. **Students at risk of school failure:** Formative and summative data will be used to monitor student success, and students at risk of failure will receive additional supports and services, which may include parent consultation, extra help, and differentiation strategies, including small group instruction, group work, scaffolding, and spiraling. **Gifted and Talented Students:** Students who excel in their mastery of course standards will be further challenged with more complex tasks, extensions of concepts and skills, and extended problem solving and critical thinking opportunities.

AP Computer Science A: This course explores the object-oriented programming paradigm using the Java language, with a concentration on problem-solving and algorithm development. This course will follow the outline recommended by the AP® Computer Science Curriculum, Exam A. Concepts such as classes, objects, inheritance, polymorphism, and code reusability are studied. It also includes algorithms, data structures, and data abstraction.

Chapter in Horstmann Textbook/ Pacing Guide (Weeks)	**Key Learning Items/Concepts**	**NJSLS CS & Design Thinking Standards**	**Formative, Summative, Benchmark, and Alternative Assessments**	**Core Instructional and Supplemental Materials/ Modifications and Accommodations**
Unit 3 10 (20) Testing and Debugging 11 (21-22) Interfaces and Polymorphism 12 (23-24) Case Study Part 2-3 Celebrity Lab Steganography 13 (25-26) Inheritance 14 (27) Case Study Part 4 15 (28) Exception Handling 17 (29) Object-Oriented Design	**Week 20:** • Carry out unit tests • Understand the principles of test case selection and evaluation • Learn strategies for effective debugging • Testing and Debugging **Weeks 21-22:** • Using interfaces for code reuse • Converting between class and interface types • Understand the concept of Polymorphism • Appreciate how interfaces can be used to decouple classes. • Abstract classes **Weeks 23-24:** • Experimenting with a large program **Weeks 25-27:** • Inheritance **Week 28:** • Understand the exceptions that occur when their programs contain errors **Week 29:** • Encapsulation • Is-a and has-a inheritance relationships • Identify inheritance, aggregation, and	**NJSLS Content Standards** 8.1.12.AP.1 8.1.12.AP.2 8.1.12.AP.3 8.1.12.AP.4 8.1.12.AP.5 8.1.12.AP.6 8.1.12.AP.7 8.1.12.AP.8 8.1.12.AP.9 8.1.12.CS.4 8.1.12.DA.2 8.1.12.DA.5 **NJSLS SMP** MP1. Make sense of problems and persevere in solving them MP2. Construct viable arguments and critique the reasoning of others MP3. Reason abstractly and quantitatively MP4. Model with mathematics MP5. Attend to precision MP6. Use appropriate tools strategically MP7. Look for and make use of structure MP8. Look for and express regularity in repeated reasoning	*Suggestion(s):* Students will be assessed regularly throughout this course, with a focus on both conceptual understanding and procedural fluency. Assessment tools may include the following: - quizzes (F) - tests (S) - performance tasks (F/S) - projects (S) - homework (F) - discussions (F) - journals (F) - Form A, B, or C benchmark (B) - alternative assessments (A) - portfolio (F, S) - online learning courses (F) - Group Learning Projects (F) - Individual Projects (S) - Oral Presentations (S)	Selection of primary sources *Suggestion(s):* College Board. *AP GridWorld Case Study.* New York: College Entrance Examination Board, 2006. (on grade level); Horstmann, Cay. *Java Concepts*. 4th ed. Hoboken, N.J.: Wiley, 2006. (on grade level; remediation); Trees, Frances P. *AP Computer Science Study Guide to Accompany Java Concepts*. 4th ed. Hoboken, N.J.: Wiley, 2006. (on grade level; remediation; advanced) **Programs:** P11.2 –P11.5(Die Class w/ Measurable Interface) Group Project11.2: Design a general program for managing board games with 2 players. Write algorithm and choose appropriate data structures. P13.3,P13.4 Project 17.2 (Design and implement Tic-Tac-Toe) **Modifications and Accommodations:** **Students with special needs:** Teachers and support staff will attend to all modifications and accommodations listed in students' IEPs and 504s. Teachers will incorporate manipulatives, extra time, alternative assessments, scaffolding, spiraling, technology, and flexible grouping to support student learning.

	dependency relationships between classes • Learn about the Software life cycle • Design and implement a class according to given specifications **Content-specific modifications and accommodations** - use multiple representations and technology to support conceptual understanding - provide students with skeletons of code and/or utilize flexible grouping **Interdisciplinary/additional connections** - Engage in programming assignments that have applications in science, mathematics, or business - Consider potential programming applications for climate change - Draw on contexts from diverse groups for programming exercises	1. Fostering an inclusive Computing Culture 2. Collaborating around Computing 3. Recognizing and Defining Computational Problems 7. Communicating about Computing **NJSLS for ELA Companion Standards** RST.9-10.3 RST.9-10.4 RST.9-10.7 RST.11-12.3 RST.11-12.4 **NJSLS-CLKS** **- 21st Century Life and Careers** 9.4.12.CI.1 9.4.12.CI.3 9.4.12.CT.1 9.4.12.CT.2 **- Technology** 9.4.12.IML.3 **- Career Education** 9.2.12.CAP.2 9.3.ST.5 9.3.ST-ET.5 9.3.ST-SM.2		**Multilingual students:** Teachers and support staff will work to support multilingual students in their first language and in English, providing materials and/or resources to support students' understanding. Students will be given additional time, as appropriate, and translation tools will be utilized as needed. **Students at risk of school failure:** Formative and summative data will be used to monitor student success, and students at risk of failure will receive additional supports and services, which may include parent consultation, extra help, and differentiation strategies, including small group instruction, group work, scaffolding, and spiraling. **Gifted and Talented Students:** Students who excel in their mastery of course standards will be further challenged with more complex tasks, extensions of concepts and skills, and extended problem solving and critical thinking opportunities.

AP Computer Science A: This course explores the object-oriented programming paradigm using the Java language, with a concentration on problem-solving and algorithm development. This course will follow the outline recommended by the AP® Computer Science Curriculum, Exam A. Concepts such as classes, objects, inheritance, polymorphism, and code reusability are studied. It also includes algorithms, data structures, and data abstraction.

Chapter in Horstmann Textbook/ Pacing Guide (Weeks)	**Key Learning Items/Concepts**	**NJSLS CS & Design Thinking Standards**	**Formative, Summative, Benchmark, and Alternative Assessments**	**Core Instructional and Supplemental Materials/ Modifications and Accommodations**
Unit 4 18 (30) Recursion 19 (31-32) Sorting and Searching (33-end) Review/ AP with WE Service-learning	**Week 30:** • Thinking Recursively • Permutations • Tracing Through Recursive Methods • Analyze problems that are much easier to solve by recursion than by iteration. • Understand when the use of recursion affects the efficiency of an algorithm. **Weeks 31-32:** • Study several sorting and searching algorithms. • Traversals, Insertions and Deletions • Selection Sort • Insertion Sort • Merge Sort • Sequential Search • Binary Search • Learn how to estimate and compare the efficiency of algorithms. **Week 33-35:** • Review AP Computer Science A topics. **Week 36-End:** • Work on Final Project Part 1: Investigate and Learn	**NJSLS Content Standards** 8.1.12.DA.5 8.1.12.DA.6 8.1.12.AP.1 8.1.12.AP.2 8.1.12.AP.4 8.1.12.AP.5 8.1.12.AP.6 8.1.12.AP.7 8.1.12.AP.9 8.2.12.NT.1 8.2.12.EC.2 8.2.12.EC.3 8.1.12.IC.1 8.1.12.IC.3 **NJSLS SMP** MP1. Make sense of problems and persevere in solving them MP2. Construct viable arguments and critique the reasoning of others MP3. Reason abstractly and quantitatively MP4. Model with mathematics MP5. Attend to precision MP6. Use appropriate tools strategically MP7. Look for and make use of structure MP8. Look for and express regularity in repeated reasoning 1. Fostering an inclusive Computing Culture	*Suggestion(s):* Students will be assessed regularly throughout this course, with a focus on both conceptual understanding and procedural fluency. Assessment tools may include the following: - quizzes (F) - tests (S) - performance tasks (F/S) - projects (S) - homework (F) - discussions (F) - journals (F) - Form A, B, or C benchmark (B) - alternative assessments (A) - portfolio (F, S) - online learning courses (F) - Group Learning Projects (F) - Individual Projects (S) - Oral Presentations (S) - Final Project (S) - Presentations (S) - Service-learning Project (S)	Selection of primary sources *Suggestion(s):* College Board. *AP GridWorld Case Study.* New York: College Entrance Examination Board, 2006. (on grade level); Horstmann, Cay. *Java Concepts*. 4th ed. Hoboken, N.J.: Wiley, 2006. (on grade level; remediation); Trees, Frances P. *AP Computer Science Study Guide to Accompany Java Concepts*. 4th ed. Hoboken, N.J.: Wiley, 2006. (on grade level; remediation; advanced) **Programs:** P18.1, Towers of Hanoi Activity P19.1-19.4, 19.11 Service-learning project Final Project **Modifications and Accommodations:** **Students with special needs:** Teachers and support staff will attend to all modifications and accommodations listed in students' IEPs and 504s. Teachers will incorporate manipulatives, extra time, alternative assessments, scaffolding, spiraling, technology, and flexible grouping to support student learning. **Multilingual students:** Teachers and support staff will work to support multilingual students in their first language and in English, providing materials and/or resources to support students'

	Part 2: Action Plan Part 3: Take Action Part 4: Report & Celebrate **Content-specific modifications and accommodations** - use multiple representations and technology to support conceptual understanding - provide students with skeletons of code and/or utilize flexible grouping **Interdisciplinary/additional connections** - Engage in programming assignments that have applications in science, mathematics, or business - Consider potential programming applications for climate change - Draw on contexts from diverse groups for programming exercises	2. Collaborating around Computing 3. Recognizing and Defining Computational Problems 7. Communicating about Computing **NJSLS for ELA Companion Standards** RST.9-10.3 RST.9-10.4 RST.9-10.7 RST.11-12.3 RST.11-12.4 **NJSLS-CLKS** **- 21st Century Life and Careers** 9.4.12.CI.1 9.4.12.CI.3 9.4.12.CT.1 9.4.12.CT.2 **- Technology** 9.4.12.IML.3 **- Career Education** 9.2.12.CAP.2 9.3.ST.5 9.3.ST-ET.5 9.3.ST-SM.2		understanding. Students will be given additional time, as appropriate, and translation tools will be utilized as needed. **Students at risk of school failure:** Formative and summative data will be used to monitor student success, and students at risk of failure will receive additional supports and services, which may include parent consultation, extra help, and differentiation strategies, including small group instruction, group work, scaffolding, and spiraling. **Gifted and Talented Students:** Students who excel in their mastery of course standards will be further challenged with more complex tasks, extensions of concepts and skills, and extended problem solving and critical thinking opportunities.

Examples of Teaching Strategies

Final Project and Student Presentations

Students are required to do a 20-30 minute presentation in June on their final project. There are three parts to the final project. Students are required to research a topic not taught in the AP A curriculum, such as advanced data structures. They are to produce a paper on that topic. Then teach the class the topic using power-point presentations and examples. A large-scale group programming project is the third component. The presentation will also detail the design and implementation of the large-scale program that students have proposed, planned, designed, and written. After each presentation, time is allotted for questions from me and other students in the class.

Lab Component

I give at least one program per chapter, and students work on their programs about 60 percent of the time. Each student has a laptop with the necessary software installed.
Students can work on programs at home and in class. They work independently and with a partner when appropriate.