COURSE TITLE

AP Computer Science—Java

LENGTH

Full Year
Grades: 11-12

DEPARTMENT

Computer Department
Barbara O'Donnell, Supervisor

SCHOOL

Rutherford High School

DATE

September 10, 2018

Initial BOE Approval Date (Born on): 5/11/2015

AP COMPUTER SCIENCE

I. Introduction/Overview/Philosophy

The major emphasis in the AP Computer Science course is on programming methodology, algorithms, and data structures. Applications provide the context in which these subjects are treated. The programming language (JAVA) is the vehicle for implementing computer-based solutions to particular problems and projects. The course is highly symbolic and demands strong problem solving skills. This course serves as an introduction for students who are planning on majoring in computer science in college and those who will major in disciplines requiring significant involvement. The course will prepare students for the AP Computer Science A exam.

The course prerequisites are the successful completion of: Computer Programming1, Computer Programming C++, and Introduction to Java.

II. Objectives

Course Outline:

I. Review of Computer Systems
- A. Introduction
  1. Overview of basic computer processing
  2. Types of software including system programs and application programs
  3. Analog information vs. digital technology
  4. Binary values and digital electronic signals
- B. Anatomy of a Computer
  1. Basic computer architecture including CPU, main memory and I/O devices
  2. Memory location, storage capacity, ROM and RAM
- C. Networks
  1. A simple computer network with printer and file servers
  2. Network connections such as point-to-point connections, LAN and WAN
  3. Internetworking and TCP/IP software
  4. The World Wide Web and HTML
- D. Programming
  1. Purpose of programming in general
  2. Introduction of the Java programming language
  3. Simple, complete Java programs with inline documentation, identifier reserved words, and white space
- E. Programming Languages
  1. Historical development of computer languages
  2. Low-level vs. high-level languages
  3. Evolution of the Java language.
  4. Basic software tools including editor, compiler, and interpreter
  5. Syntax rules and the semantics of a language
  6. Compile-time error vs. runtime error

II. Review of Objects and Primitive Data
- A. An Introduction to Objects
  1. Primitive data, variables, objects, classes, inheritance, and encapsulation
  2. Aspects of object-oriented software
- B. Using Objects
  1. The print and println methods
  2. Abstraction and levels of abstraction
- C. String Literals
  1. String concatenation

        2. Escape sequences
    D. Variables and Assignment
        1. Declaration and use of variables in a program
        2. The assignment statement
        3. Constants
    E. Primitive Data Types
        1. Integers and floating points
        2. Booleans and Boolean literals
        3. Characters, character set and ASCII
    F. Arithmetic Expressions
        1. Operator precedence, unary and binary operators
        2. Data conversion and arithmetic promotion
    G. Creating Objects
        1. Reference to an object
        2. The string class, method header, return type
        3. The wrapper classes
    H. Class Libraries and Packages
        1. The Java standard class library, API documentation
        2. Organization of classes in packages
        3. The import declaration
        4. The random class
    I. Invoking Class Methods
        1. Class methods vs. static methods
        2. The math class, the scanner class
    J. Formatting Output
        1. The NumberFormat class
        2. The DecimalFormat class

III. Review of Program Statements
    A. Program Development
        1. The four basic development activities
        2. Software design
    B. Control Flow
        1. Conditional and selection statements
        2. Repetition statements and three types of loops
    C. The If Statement
        1. The logic of an if statement
        2. The equality and relational operators
        3. The if-else statement
        4. Using block statements
        5. Nested if statements

    D. Boolean Expressions Revisited
        1. Logical operators, OR, AND
        2. Comparing characters and strings
        3. Comparing floating point values
    E. More Operators
        1. Increment and decrement operators
        2. Assignment operators
    F. The While Statement
        1. The logic of a while loop
        2. Infinite loops
        3. Nested loops
        4. The StringTokenizer class and its methods

- G. The For Statement
  1. The logic of a for loop
  2. Comparing loops
- H. Program Development Revisited
  1. Requirements
  2. Design questions
  3. Sketch out algorithm
  4. Implementation of algorithm

IV. Implementing Classes

- A. Objects Revisited
  1. Object's state and behavior
  2. Classes
- B. Anatomy of a Class
  1. Data and method declarations
  2. Instance data
  3. Class definition
  4. Specifying and commenting the public instance of a class
- C. Anatomy of a Method
  1. The flow of control following method invocations
  2. Method definition
  3. Tester programs
  4. The return statement
  5. Implicit and explicit method parameters
  6. Constructor definition
  7. Local data
- D. Categories of Variables
  1. Access specifier and type
  2. Instance field definition
  3. Local variables
  4. Parameter variables
- E. Implementing Constructors and Methods
  1. Supply bodies of the constructors and methods to a class
  2. Guidelines to implementing a class
  3. Unit testing

V. Enhancing Classes

- A. References Revisited
  1. The null reference
  2. The this reference
  3. Aliases
  4. Passing objects as parameters
- B. The static modifier
  1. Static variables
  2. Static methods
  3. Reference to math class
- C. Exceptions
  1. Exception messages
  2. Throwing exceptions
- D. Interfaces
  1. Abstract methods
  2. Implementing an interface
  3. The Comparable interface
  4. The List interface

5. The Iterator and ListIterator interfaces

E. Designing Classes

1. State of the object
2. Behavior of the object

VI. Arrays

A. Format of Arrays

1. Array indexing
2. Declaring and using arrays
3. Automatic bounds checking
4. Off-by-one-error
5. Initializer lists
6. Arrays parameters

B. Arrays of Objects

1. Arrays of String objects
2. Command-line arguments
3. Filling arrays of objects
4. Copying arrays

C. Searching

1. Linear or sequential search
2. Binary search

D. Sorting

1. Selection sort
2. Insertion sort
3. Sorting an array of objects

E. Comparing Sorts

1. Time efficiency vs. space efficiency
2. The Big-Oh

H. The ArrayList class

1. Some methods of the ArrayList class using Java 5
2. Using a ListIterator
3. ArrayList Efficency
4. Enhanced for loop

VII. Inheritance

A. Creating Subclasses

1. Derived classes
2. Child and parent classes
3. Is-A relationship
4. The super reference
5. Multiple inheritance

B. Overriding Methods

1. In inheritance

C. Class Hierarchies

1. Siblings
2. The Object class
3. The Abstract classes

D. Indirect Use of Class Members

1. Members of a superclass from a subclass
2. Inherited members and how they are referenced

E. Polymorphism

1. Polymorphic reference
2. Polymorphism and inheritance

F. Interfaces

1. Polymorphism with interfaces
2. Using interfaces for code reuse
3. Defining and implementing an interface

VIII. Recursion

A. Recursive Thinking

1. Infinite recursion
2. Recursion in math
3. Permutations

B. Recursive Programming

1. Recursion vs. iteration
2. Direct vs. indirect recursion
3. Efficiency of recursion

C. Using Recursion

1. Triangle Numbers
2. Fibonacci Sequence
3. Towers of Hanoi

D. Recursion in Sorting

1. Merge Sort
2. Quick Sort

IX. Greenfoot

A. Review of basic Greenfoot concepts

1. Creating a World with classes and objects
2. Making objects act
4. Invoking methods
5. Interacting with other objects
6. Playing sounds
7. Movement and key control

B. Advanced techniques

1. Explosions
2. Animated actors
3. Shooting
4. Displaying Texts
5. Counters
6. Timers
7. Color Masks
8. Scrolling
9. Animated Gifs

X. The Lab Component

The AP CS A Exam requires a minimum of 20 hours hands-on lab experiences. The labs provided by the Collegeboard are not required. The Labs are:

1. The Elevens Lab
2. The Magpie Lab
3. The Picture Lab

Student Outcomes:

Upon completion of the course, students will demonstrate the ability to:

- Describe the relationship between hardware and software.
- Define various types of software and how they are used.
- Identify basic computer hardware and explain what it does.
- Explain how the hardware components execute programs and manage data.
- Describe how computers are connected into networks to share information.
- Explain the importance of the Internet and the World Wide Web.

- Introduce the Java programming language.
- Describe the steps involved in program compilation and execution.
- Define the difference between primitive data and objects.
- Declare and use variables.
- Perform mathematical computations.
- Create objects and use them.
- Discuss basic program development steps.
- Define the flow of control through a program.
- Learn to use if statements.
- Define expressions that let us make complex decisions.
- Learn to use while and for statements.
- Define classes that act like blueprints for new objects, made of variables and methods.
- Explain encapsulation and Java modifiers.
- Explore the details of method declarations.
- Review method invocation and parameter passing.
- Explain and use method overloading.
- Learn to divide complicated methods into simpler, supporting methods.
- Describe relationships between objects.
- Define reference aliases.
- Explore passing object references as parameters.
- Learn to use the static modifier.
- Define nested classes and inner classes.
- Define and use arrays.
- Describe how arrays and array elements are passed as parameters.
- Explore how arrays and other objects can be combined to manage complex information.
- Explore searching and sorting with arrays.
- Learn to use multidimensional arrays.
- Examine the ArrayList class.
- Derive new classes from existing ones.
- Explain how inheritance supports software reuse.
- Add and modify methods in child classes.
- Discuss how to design class hierarchies.
- Define polymorphism and how it can be done.
- Explain the underlying ideas of recursion.
- Examine recursive methods and processing steps.
- Define infinite recursion and discuss ways to avoid it.
- Explain when recursion should and should not be used.
- Demonstrate the use of recursion to solve problems.
- Examine the use of recursion in sorting.
- Read and understand a large program consisting of several classes and interacting objects, as well as the description of the design and development process leading to such a program.
- Review basic concepts of Greenfoot.
- Use advanced techniques of Greenfoot to develop interactive games and simulations.

NEW JERSEY STUDENT LEARNING STANDARDS

CAREER READY PRACTICES

CRP1 Act as a responsible and contributing citizen and employee

Career-ready individuals understand the obligations and responsibilities of being a member of a community, and they demonstrate this understanding every day through their interactions with others.

They are conscientious of the impacts of their decisions on others and the environment around them. They think about the near-term and long-term consequences of their actions and seek to act in ways that contribute to the betterment of their teams, families, community and workplace. They are reliable and consistent in going beyond the minimum expectation and in participating in activities that serve the greater good.

CRP2 Apply appropriate academic and technical skills

Career-ready individuals readily access and use the knowledge and skills acquired through experience and education to be more productive. They make connections between abstract concepts with real-world applications, and they make correct insights about when it is appropriate to apply the use of an academic skill in a workplace situation.

CRP4 Communicate clearly and effectively and with reason.

Career-ready individuals communicate thoughts, ideas, and action plans with clarity, whether using written, verbal, and/or visual methods. They communicate in the workplace with clarity and purpose to make maximum use of their own and others' time. They are excellent writers; they master conventions, word choice, and organization, and use effective tone and presentation skills to articulate ideas. They are skilled at interacting with others; they are active listeners and speak clearly and with purpose. Career-ready individuals think about the audience for their communication and prepare accordingly to ensure the desired outcome.

CRP6 Demonstrate creativity and innovation

Career-ready individuals regularly think of ideas that solve problems in new and different ways, and they contribute those ideas in a useful and productive manner to improve their organization. They can consider unconventional ideas and suggestions as solutions to issues, tasks or problems, and they discern which ideas and suggestions will add greatest value. They seek new methods, practices, and ideas from a variety of sources and seek to apply those ideas to their own workplace. They take action on their ideas and understand how to bring innovation to an organization.

CRP8 Utilize critical thinking to make sense of problems and persevere in solving them

Career-ready individuals readily recognize problems in the workplace, understand the nature of the problem, and devise effective plans to solve the problem. They are aware of problems when they occur and take action quickly to address the problem; they thoughtfully investigate the root cause of the problem prior to introducing solutions. They carefully consider the options to solve the problem. Once a solution is agreed upon, they follow through to ensure the problem is solved, whether through their own actions or the actions of others.

CRP11 Use technology to enhance productivity

Career-ready individuals find and maximize the productive value of existing and new technology to accomplish workplace tasks and solve workplace problems. They are flexible and adaptive in acquiring new technology. They are proficient with ubiquitous technology applications. They understand the inherent risks-personal and organizational-of technology applications, and they take actions to prevent or mitigate these risks.

TECHNOLOGY STANDARDS

STANDARD 8.1: EDUCATIONAL TECHNOLOGY: ALL STUDENTS WILL USE DIGITAL TOOLS TO ACCESS, MANAGE, EVALUATE, AND SYNTHESIZE INFORMATION IN ORDER TO SOLVE PROBLEMS INDIVIDUALLY AND COLLABORATE AND TO CREATE AND COMMUNICATE KNOWLEDGE.

8.1.12.B.2 - Apply previous content knowledge by creating and piloting a digital learning game or tutorial.

8.1.12.D.5 - Analyze the capabilities and limitations of current and emerging technology resources and assess their potential to address personal, social, lifelong learning, and career needs.
8.1.12.F.1 - Evaluate the strengths and limitations of emerging technologies and their impact on educational, career, personal and or social needs.

STANDARD 8.2: TECHNOLOGY EDUCATION, ENGINEERING, DESIGN, AND COMPUTATIONAL THINKING – PROGRAMMING: ALL STUDENTS WILL DEVELOP AN UNDERSTANDING OF THE NATURE AND IMPACT OF TECHNOLOGY, ENGINEERING, TECHNOLOGICAL DESIGN, COMPUTATIONAL THINKING, AND THE DESIGNED WORLD AS THEY RELATE TO THE INDIVIDUAL, GLOBAL SOCIETY, AND THE ENVIRONMENT.
8.2.12.E.3 - Use a programming language to solve problems or accomplish a task (e.g., robotic functions, website designs, applications, and games).
8.2.12.E.4 - Use appropriate terms in conversation (e.g., troubleshooting, peripherals, diagnostic software, GUI, abstraction, variables, data types and conditional statements).

STANDARD 9.3: CAREER AND TECHNICAL EDUCATION
PATHWAY: PROGRAMMING & SOFTWARE DEVELOPMENT (IT-PRG)
9.3.IT-PRG.3 Analyze system and software requirements to ensure maximum operating efficiency.
9.3.IT-PRG.4 Demonstrate the effective use of software development tools to develop software applications.
9.3.IT-PRG.5 Apply an appropriate software development process to design a software application.
9.3.IT-PRG.6 Program a computer application using the appropriate programming language.
9.3.IT-PRG.7 Demonstrate software testing procedures to ensure quality products.
9.3.IT-PRG.8 Perform quality assurance tasks as part of the software development cycle.

III. Proficiency Levels

This course is open to grades 11-12.

IV. Methods of Assessment

Student Assessment

The teacher will provide a variety of assessments during the course of the year. Among these are: homework, laboratory exercises, teacher-made tests and quizzes, and long-term projects.

Curriculum/Teacher Assessment
The teacher will provide the subject area supervisor with suggestions for changes on an ongoing basis.

V. Grouping

The prerequisite for AP Computer Science is successful completion of Computer Programming I, Computer Programming C++, and Introduction to JAVA.

VI. Articulation/Scope & Sequence/Time Frame

Course length is a full-year and is offered to students in grades 11 and 12.

VII. Resources

Texts/Supplemental Reading/References

A+ Complete Currriculum Materials for A+ Computer Science (2017-2018 license)
Horstmann, Cay Java Concepts, Early Objects, Seventh Edition, Hoboken, NJ: John Wiley & Sons, Inc., 2014.
Horstmann, Cay Java Concepts, Fifth Edition, Hoboken, NJ: John Wiley & Sons, Inc., 2008.
Horstmann, Cay Java Concepts, AP Edition, Hoboken, NJ: John Wiley & Sons, Inc., 2005.
Lewis, John, Loftus, William, and Cocking, Cara JAVA Software Solutions for AP Computer Science

Pearson Education, Inc. 2004
Lewis, John, Loftus, William, and Cocking, Cara JAVA Software Solutions for AP Computer Science A, 2nd edition, Boston, MA: Addison Wesley, 2006
Schram, Leon Multiple Choice & Free Response Questions in Preparation for the AP Computer Science Examination, Ninth Edition, D & S Marketing Systems, Inc, 2015.
Schram, Leon GridWorld Case Study Manual for the AP Computer Science Examination, D & S Marketing Systems, Inc, 2013.
Introduction to Programming in Greenfoot, Kolling, Pearson, 2009
AP Computer Science A Test Preparation, Fifth Edition, Hauppage, NY: Barron's, 2010.
www.WileyPlus.com
www.codingBat.com
www.joyOfCode.org
www.greenfoot.org

VIII. Suggested Activities

- Laboratory programming problems
- Class presentations
- Cooperative programming projects

IX. Methodologies

Much of the class time is spent in lab work and on programming problems to be completed. When group instruction is necessary, topics are taught using the computer projection system in conjunction with student classwork.

X. Interdisciplinary Connections

Connections are made to mathematics by using a variety of arithmetic formulas, as well as higher mathematical concepts.

XI. Differentiating Instruction for Students with Special Needs: Students with Disabilities, Students at Risk, English Language Learners, and Gifted & Talented Students

Differentiating instruction is a flexible process that includes the planning and design of instruction, how that instruction is delivered, and how student progress is measured. Teachers recognize that students can learn in multiple ways as they celebrate students' prior knowledge. By providing appropriately challenging learning, teachers can maximize success for all students.

Differentiating in this course includes but is not limited to:

Differentiation for Support (ELL, Special Education, Students at Risk)

- suggest a timeline and schedule for students and monitor students' progress
- clarify the requirements and prompts in each performance task when students do not understand the directions
- assist students in defining their focus and choice of topics prior to them beginning their investigation without making selections for them (e.g. by asking questions)
- review the files submitted to ensure the files are correct and not corrupted
- advance organizers
- reteach and review
- online video review
- study guides
- peer mentoring on lab
- guided notetaking
- powerpoint reference

- visual demonstrations, illustrations, and models
- work with checklists

Differentiation for Enrichment

- more complex tasks and problems
- higher expectations in assessment questioning
- independent extensions based on student interest, curiosity, and choice
- extended research and readings
- peer teaching and support
- display and explain version of coded task
- facilitating classmates
- higher expectation for writing programs
- flexible grouping on challenging exercises
- higher level questioning techniques

XII. Professional Development

The teacher will continue to improve expertise through participation in a variety of professional development opportunities.

XIII. Curriculum Map/Pacing Guide

Unit Topic	Time Allocated	Differentiating Instruction for Students with Disabilities, Students at Risk of School Failure, English Language Learners, & Gifted & Talented Students	Standards	Assessments
Review of Computer Systems and Intro to Java • Anatomy of Computer • Networking • Languages • Ethics • Careers	1 Week	*For Support:* • assist students in defining their focus and choice of topics prior to them beginning their investigation without making selections for them • advance organizers • suggest a timeline and schedule for students and monitor students' progress *For Enhancement:* • independent extensions based on student interest, curiosity, and choice • extend research and readings • facilitating classmates	*Standards:* CRP1, CRP2, CRP4, 8.1.12.D.5, 8.1.12.F.1, 9.3.IT-PRG.3, 9.3.IT-PRG.4	*Formative Assessment:* • Research Majors in Computer Science and College Requirements • Worksheet (hardware vs software) *Summative Assessment:* • Powerpoint Presentation (Career in Computer Science)

Unit Topic	Time Allocated	Differentiating Instruction for Students with Disabilities, Students at Risk of School Failure, English Language Learners, & Gifted & Talented Students	Standards	Assessments
Review of Objects, Primitive Data and Program Statements • Java as an OOP language • Using Objects • String Literals • Variables and Assignment • Primitive Data Types • Arithmetic Expressions • Creating Objects • Class Libraries and Packages • Invoking Class Methods • Formatting Output • Program Development • Control Flow • The If Statement • Boolean Expressions Revisited • More Operators • The While Statement • The For Statement • Program Development Revisited	2 Weeks	*For Support:* • reteach and review • online video review • study guides *For Enhancement:* • display and explain version of coded task • facilitating classmates • higher expectation for writing programs	*Standards:* CRP6, CRP8, 8.2.12.E.3, 8.2.12.E.4	*Formative Assessment:* • Summer Assignment Review *Summative Assessment:* • Lab#1 (OOP - Basics) • Summer Assignment Test • ChatBot Lab

Unit Topic	Time Allocated	Differentiating Instruction for Students with Disabilities, Students at Risk of School Failure, English Language Learners, & Gifted & Talented Students	Standards	Assessments
Arrays • Format • Arrays of String Objects • Arrays of Class Objects • Searching • Enhanced For Loop • Manipulating Arrays	3 Weeks	*For Support:* • peer mentoring on lab • guided notetaking • powerpoint reference *For Enhancement:* • flexible grouping on challenging exercises • higher level questioning techniques • higher expectations in assessment questioning	*Standards:* CRP2, CRP6, 8.2.12.E.4, 9.3.IT-PRG.5, 9.3.IT-PRG.8	*Formative Assessment:* • Questions of the Day using Google Classroom - based on current or previously learned content • Worksheets (Tracing Arrays)(4) • Worksheets (Coding Arrays)(4) • Multiple Choice Questions (Arrays) *Summative Assessment:* • Lab#2 (Arrays) • Test (Arrays) • Collegeboard Free Response (Arrays)

Unit Topic	Time Allocated	Differentiating Instruction for Students with Disabilities, Students at Risk of School Failure, English Language Learners, & Gifted & Talented Students	Standards	Assessments
Implementing Classes • Objects Revisited • Anatomy of a Class • Anatomy of a Method • Categories of Variables • Implementing Constructors and Methods	3 weeks	*For Support:* • appeal to diverse learning styles • choice to work with others or alone • visual demonstrations, illustrations, and models *For Enhancement:* • peer teaching and support • display and explain version of coded task • facilitating classmates	*Standards:* CRP6, CRP11, 8.2.12.E.3, 9.3.IT-PRG.8	*Formative Assessment:* • Questions of the Day using Google Classroom - based on current or previously learned content • Q&A (Inheritance Video) • Class Writing Applications • Worksheets (Inheritance) (4) *Summative Assessment:* • Lab#3 (Inheritance) • Test (Inheritance)

Unit Topic	Time Allocated	Differentiating Instruction for Students with Disabilities, Students at Risk of School Failure, English Language Learners, & Gifted & Talented Students	Standards	Assessments
Enhancing Classes • References Revisited • The static modifier • Exceptions • Interfaces • Abstract Classes	3 weeks	*For Support:* • study guides • suggest a timeline and schedule for students and monitor students' progress • peer mentoring on lab *For Enhancement:* • independent extensions based on student interest, curiosity, and choice • extended research and readings • higher expectation for writing programs	*Standards:* CRP4, CRP6, CRP8, 8.2.12.E.3, 8.2.12.E.4, 9.3.IT-PRG.7	*Formative Assessment:* • Questions of the Day using Google Classroom - based on current or previously learned content • Writing abstract classes • Worksheets (interfaces and abstract classes) • Fixing code (runtime error exceptions) • Review of static objects / classes • Multiple Choice Questions (Classes) • *Summative Assessment:* • Lab#4 (abstract classes) • Test (interfaces and abstract classes) • Benchmark Q2

Unit Topic	Time Allocated	Differentiating Instruction for Students with Disabilities, Students at Risk of School Failure, English Language Learners, & Gifted & Talented Students	Standards	Assessments
ArrayList • Wrappers • Basics • Methods • Using objects • Static Methods • Enhanced For Loop	3 weeks	*For Support:* • review the files submitted to ensure the files are correct and not corrupted • visual demonstrations, illustrations, and models • work with checklists *For Enhancement:* • peer teaching and support • display and explain version of coded task • facilitating classmates	*Standards:* CRP2, CRP6, CRP8, 8.1.12.B.2, 8.2.12.E3, 9.3.IT-PRG.7, 9.3.IT-PRG.8	*Formative Assessment:* • Questions of the Day using Google Classroom - based on current or previously learned content • Worksheets (ArrayLists) (2) • Multiple Choice Questions (ArrayLists) • Worksheets (Coding using ArrayLists) (2) • Collegeboard Free Response (ArrayLists) *Summative Assessment:* • Lab#5 (ArrayLists) • Test (ArrayLists)

Unit Topic	Time Allocated	Differentiating Instruction for Students with Disabilities, Students at Risk of School Failure, English Language Learners, & Gifted & Talented Students	Standards	Assessments
Inheritance • Creating Subclasses • Overriding Methods • Class Hierarchies • Indirect Use of Class Members • Polymorphism • Interfaces Revisited	6 weeks	*For Support:* • assist students in defining their focus and choice of topics prior to them beginning their investigation without making selections for them • online video review • peer mentoring on lab *For Enhancement:* • peer teaching and support • more complex tasks and problems • higher expectations in assessment questioning	*Standards:* CRP6, CRP8, CRP11, 8.1.12.B.2, 8.2.12.E.4, 9.3.IT-PRG.5	*Formative Assessment:* • Questions of the Day using Google Classroom - based on current or previously learned content • Worksheets (Inheritance) (3) • Worksheets (Polymorhism) (4) • Multiple Choice Questions) (Inheritance / Polymorphism) • Collegeboard Free Response (Designing Classes) *Summative Assessment:* • Quiz (Inheritance) • Elevens Lab • Lab#5 (Polymorphism) • Test (Inheritance / Polymorphism)

Unit Topic	Time Allocated	Differentiating Instruction for Students with Disabilities, Students at Risk of School Failure, English Language Learners, & Gifted & Talented Students	Standards	Assessments
Two Dimensional Arrays • Declaring • Accessing Elements • Locating Neighboring Elements • Accessing Rows and Columns • Manipulating Two Dim Arrays	3 weeks	*For Support:* • clarify the requirements and prompts in each performance task when students do not understand the directions • work with checklists • peer mentoring on lab *For Enhancement:* • independent extensions based on student interest, curiosity, and choice • peer mentoring • display and explain version of coded task	*Standards:* CRP6, CRP8, 8.1.12.B.2, 8.2.12.E.3, 8.2.12.E.4, 9.3.IT-PRG.3	*Formative Assessment:* • Questions of the Day using Google Classroom - based on current or previously learned content • Worksheet (Tracing) (2) • Worksheets (Coding) (4) • Multiple Choice Questions (Matrices) • Collegeboard Free Response (TwoDim Arrays) *Summative Assessment:* • Quiz (TwoDim) • Matrix Lab • Picture Lab • Test (TwoDim) • Benchmark Q3

Unit Topic	Time Allocated	Differentiating Instruction for Students with Disabilities, Students at Risk of School Failure, English Language Learners, & Gifted & Talented Students	Standards	Assessments
Recursion • Recursive Thinking • Recursive Programming • Using Recursion • Recursion in Sorting	3 weeks	*For Support:* • clarify the requirements and prompts in each performance task when students do not understand the directions • work with checklists • peer mentoring on lab *For Enhancement:* • independent extensions based on student interest, curiosity, and choice • peer mentoring • display and explain version of coded task	*Standards:* CRP6, CRP8, CRP11, 8.2.12.E.3, 8.2.12.E.4, 9.3.IT-PRG.3, 9.3.IT-PRG.6	*Formative Assessment:* • Questions of the Day using Google Classroom - based on current or previously learned content • Worksheet (Recursion) (4) • Coding Practice from CodingBat (Recursion) • Multiple Choice Questions (Recursion) *Summative Assessment:* • Quiz (Recursion)
Searching and Sorting • Binary Search / Sort • Selection Sort • Merge Sort • Insertion Sort • Profile each Sorting Algorithm • Analyze Performance of Each Sorting Algorithm	3 weeks	*For Support:* • visual demonstrations, illustrations, and models • powerpoint reference • online video review *For Enhancement:* • flexible grouping on challenging exercises • higher level questioning techniques	*Standards:* CRP4, CRP8, 8.2.12.E.4, 9.3.IT-PRG.7, 9.3.IT-PRG.8	*Formative Assessment:* • Questions of the Day using Google Classroom - based on current or previously learned content • Worksheets (Tracing) (4) • Coding practice with each sorting method • Multiple Choice Questions (Sorting Methods) *Summative Assessment:* • Quiz (Search/Sort)

Unit Topic	Time Allocated	Differentiating Instruction for Students with Disabilities, Students at Risk of School Failure, English Language Learners, & Gifted & Talented Students	Standards	Assessments
Review for AP Exam • Cumulative Review of All Topics • Strategies	4 weeks	*For Support:* • reteach and review • study guides • suggest a timeline and schedule for students and monitor students' progress *For Enhancement:* • independent more complex tasks and problems • display and explain version of coded task • higher expe	*Standards:* CRP6, CRP8, CRP11, 8.2.12.E.3, 8.2.12.E.4, 9.3.IT-PRG.3, 9.3.IT-PRG.6	*Formative Assessment:* • Questions of the Day using Google Classroom - based on current or previously learned content • Multiple Choice Questions (All Topics) • Practice APCS Exam (Collegeboard) • Free Response (Collegeboard) *Summative Assessment:* • Test (Sample APCS Exam)

Unit Topic	Time Allocated	Differentiating Instruction for Students with Disabilities, Students at Risk of School Failure, English Language Learners, & Gifted & Talented Students	Standards	Assessments
Greenfoot • Review of basic Greenfoot concepts o Advanced techniques o Explosions o Animated actors o Shooting o Displaying Texts o Counters o Timers o Color Masks o Scrolling o Animated Gifs	6 weeks	*For Support:* • suggest a timeline and schedule for students and monitor students' progress • assist students in defining their focus and choice of topics prior to them beginning their investigation without making selections for them • work with checklists *For Enhancement:* • independent extensions based on student interest, curiosity, and choice • higher expectation for writing programs • higher expectations in assessment questioning	*Standards:* CRP6, CRP8, CRP11, 8.1.12.B.2, 8.2.12.E.4, 9.3.IT-PRG.4, 9.3.IT-PRG.6	*Formative Assessment:* • Complete code for trick-the-turtle and ketchup projects • Animated Gif project • FlappyBird project progression (4) • Joy of Code (series 17-26) • Daily Progress Entries of Final Project *Summative Assessment:* • Quiz (Project Plan) • Test - Greenfoot Project (Written) • Test - Greenfoot Project Presentation • Benchmark Q4