AP COMPUTER SCIENCE
GRADES 10 – 12

THE EWING PUBLIC SCHOOLS
2099 Pennington Road
Ewing, NJ 08618

Contents

Course Description

AP Computer Science is a one-semester course that will meet for five 87-minute class periods per week. It is designed to immediately follow Introduction to Computer Science and to prepare students for the AP CS Exam. The course includes a number of individual programming projects assigned for one week each. The time after the AP CS Exam is devoted to team project and enrichment activities. The course is based on numerous problem solving exercises, labs, and case studies, which require students to design and implement Java classes. [CR1] The course requires daily access to a computer lab or 87 minutes per day on a daily basis. [CR6]

Interdisciplinary Connections

Connections to Mathematics

Students will use digital tools to explore, model, synthesize and evaluate mathematical concepts. Technology should enhance and enrich the learning experience and provide students with new and alternative ways to access mathematical content. (NJSLS Technology 8.1, MP1, 2, & 4)

Connections to Science

Many of the mathematical practices and the science and engineering practices of Next Generation Science Standards (NGSS) are interrelated, specifically, the practices of developing and using mathematical models, computational thinking, and strategic use of mathematical tools. (NJSLS Science SEP 2, 5)

Connections to English Language Arts

Students are expected to develop viable arguments and to critique the reasoning of others. This performance expectation is present in ELA and math practice standards as well as NGSS science and engineering practices. Also, inherent in all three sets of standards is an expectation to research and communicate information and enduring understandings. (NJSLSA.W1, NJSLSA.W7-9, NJSLS Science SEP 7, 8)

21st Century Standards for College and Career Readiness

Think Creatively

- o Use a wide range of idea creation techniques (such as brainstorming)
- o Create new and worthwhile ideas (both incremental and radical concepts)
- o Elaborate, refine, analyze and evaluate their own ideas in order to improve and maximize creative efforts

Work Creatively with Others

- o Develop, implement and communicate new ideas to others effectively
- o Be open and responsive to new and diverse perspectives; incorporate group input and feedback into the work
- o Demonstrate originality and inventiveness in work and understand the real world limits to adopting new ideas
- o View failure as an opportunity to learn; understand that creativity and innovation is a long-term, cyclical process of small successes and frequent mistakes

Implement Innovations

- o Act on creative ideas to make a tangible and useful contribution to the field in which the innovation will occur

Reason Effectively

- o Use various types of reasoning (inductive, deductive, etc.) as appropriate to the situation

Use Systems Thinking

- o Analyze how parts of a whole interact with each other to produce overall outcomes in complex systems

Make Judgments and Decisions

- o Effectively analyze and evaluate evidence, arguments, claims and beliefs
- o Analyze and evaluate major alternative points of view
- o Synthesize and make connections between information and arguments
- o Interpret information and draw conclusions based on the best analysis
- o Reflect critically on learning experiences and processes

Solve Problems
- o Solve different kinds of non-familiar problems in both conventional and innovative ways
- o Identify and ask significant questions that clarify various points of view and lead to better solutions

Curricular Requirements

CR1: This course teaches students to design and implement computer based solutions to problems.
See pages 2-3,6-7,9-10,13,15-16,18-20, 22-25

CR2a: The course teaches students to use and implement commonly used algorithms.
See pages 17, 20, 24

CR2b: The course teaches students to use commonly used data structures.
See pages 17, 20

CR3: The course teaches students to select appropriate algorithms and data structures to solve problems.
See pages 10, 20, 24, 25

CR4: The course teaches students to code fluently in an object-oriented paradigm using the programming language Java.
See pages 9, 12-14, 16-17, 19, 20-21, 23-24

CR5: The course teaches students to use elements of the standard Java library from the AP Java subset in Appendix A of the AP Computer Science A Course Description.
See pages 5, 6, 8, 14, 17, 20-22

CR6: The course includes a structured lab component comprised of a minimum of 20 hours of hands-on lab experiences.
See pages 6-7, 9, 10, 13, 15-16, 18-20, 22-25

CR7: The course teaches students to recognize the ethical and social implications of computer use.
See pages 26

Instructional Resources

- ❖ Lewis, Loftus and Cocking Java Software Solutions for AP Computer Science 3rd Edition or later. Addison-Weslay, 2011

- ❖ Litvin, Maria, and Gary Litvin. Be Prepared for the AP Computer Science Exam in Java, 6th Edition, Andover, Mass.: Skylight Publishing, 2014.

- ❖ The College Board's AP Computer Science A Elevens Lab Student Guide

- ❖ The College Board's AP Computer Science A Magpie Chatbot Lab Student Guide

- ❖ The College Board's AP Computer Science A Picture Lab Student Guide

- ❖ Current media sources and Internet articles and blogs discussing ethical and social issues related to computer use.

- ❖ The College Board's Computer Science A Course Description.

- ❖ The College Board's Magpie, Picture Lab, and Elevens Labs Teacher Guides.

- ❖ AP Central resources.

- ❖ Java Software Solutions student files, teacher files, PowerPoint presentations, Test Package, additional resources

- ❖ Parlante, Nick. CodingBat: http//codingbat.com/java

Programming Environment

- ❖ NetBeans IDE

- ❖ BlueJ

Unit #1: Semester 1/Weeks 1-3
Big Ideas: One-Dimensional Arrays/Two Dimensional Arrays/Searching

Text and Resources:
- Java Software Solutions for AP Computer Science 3rd Edition or later. Addison-Weslay, 2011
- Be Prepared for the AP Computer Science Exam in Java, 6th Edition, Andover, Mass.: Skylight Publishing, 2014.
- Java Software Solutions student files, teacher files, PowerPoint presentations, Test Package, additional resources
- CodingBat: http//codingbat.com/java

Content Summary:
- 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
- One- and two-dimensional arrays algorithms (insertions, deletions, traversals)
- Searching with arrays (sequential search, binary search)
- Comparison (sequential and binary)
- Choose appropriate data representation and algorithms.

Assessments:
Formative
1. Java Software Solutions, sections 6.0-6.2, 6.6; Summary of Key Concepts (redacted).
2. PictureLab Introduction and Activities 1-9 (APCS A Labs).
3. Java Software Solutions, Self-Review Questions 6.1-6.9.
4. Array Worksheets.
5. Java Software Solutions Short Answer Exercises

Summative
1. Clicker Questions – Java Software Solutions, Multiple Choice 6.1-6.5, 6.8; True/False 6.1-6.7; AP-Style Multiple Choice 6.1-6.5.
2. Multiple Choice and Free Response Exam.

Sample Activities:
- Java Software Solutions, Programming Project 6.4 – Design, implement, and test a program that inputs integers and produces a histogram. [CR1]
- Picture Lab Activities 1-9 (APCS A Labs). [CR4]
- Java Software Solutions Programming Projects 6.1: Design and implement an application that reads a number of Integers and count how many times each one is entered. Print appropriate feedback
- Write a program that keeps an appointment book. Make a class Appointment that stores a description of the appointment, the appointment day, ending

time. Your program should keep the appointments in a sorted arrayList. Users can add appointments and print out all the appointments for a given day. When a new appointment is added, use binary search to find where it should be inserted in the arrayList. Do not add it if it conflicts with another appointment. [CR4]

- Download the CD and Card classes from shared folder
- Use the CD class to implement a CD collection class that include the following methods and declarations:
- Array collection of type CD
- Integer count to keep track of the used spots in the array
- Double variable that represents the cost
- Constructor
- A method to add a CD that takes Two strings, a double and an integer as parameter
- Ensure capacity method
- Print method or toString method that allows to print the entire collection
- Add a search method that finds an specific CD (Use equals method)
- Add a countOccurrences method
- Add and insertion method and a delete method
- Use the Card class to implement the following methods:
- Add a default constructor
- Add a method that return an array of cards representing a deck of cards
- Add a method that prints an array of cards
- Add a Boolean method that fourOfaKind that returns true if a hand of 5 cards contain 4 cards with the same face value
- Implement a driver program to test the methods
- Change the methods fourOfaKind and print static
- Write a function that performs a linear search on a set of integer values. Return –1 if the value is not found.
- Analyze the efficiency of the algorithm
- Perform a Binary Search using a set of index cards
- Write the pseudo-code for the binary search
- Implement the binary search
- Use the set of cards to remove an element at the given position
- Implement the corresponding function

Standards Addressed:

[CR1]: This course teaches students to design and implement computer based solutions to problems

[CR4]: The course teaches students to code fluently in an object-oriented paradigm using the programming language Java.

Unit #2: Semester 1/Weeks 4-6
Big Ideas: Lists/ ArrayLists/ Selection and Insertion Sort
Text and Resources:
- Java Software Solutions for AP Computer Science 3rd Edition or later. Addison-Weslay, 2011
- Be Prepared for the AP Computer Science Exam in Java, 6th Edition, Andover, Mass.: Skylight Publishing, 2014.
- Java Software Solutions student files, teacher files, PowerPoint presentations, Test Package, additional resources
- CodingBat: http//codingbat.com/java

Content Summary:
- Lists / ArrayLists
- Selection and Insertion Sorts [CR2a]
- Lists and ArrayLists (creation, insertions, deletions, traversals, algorithms); [CR2b] [CR5]
- Sorting algorithms and comparison (selection and insertion) [CR2a] [CR3]
- Choosing appropriate data representation and algorithms [CR3]

Assessments:
Formative
> 1. Java Software Solutions, sections 6.3-6.4, 6.7; Summary of Key Concepts (redacted).
> 2. Elevens Activities 2-4 (APCS A Labs).
> 3. Java Software Solutions, Self-Review Questions 6.10, 6.12.
> 4. Java Software Solutions Short-Answer Exercises

Summative
> 1. Java Software Solutions Chapter 6 AP-Style Multiple Choice and Free Response Questions
> 2. Multiple Choice and Free Response Exam.

Sample Activities:
- Elevens Activities 2-4 – Deck Class (APCS A Labs). [CR4]
- Design and implement an application that computes and prints the mean and standard deviation of a list of integers.
- Java Software Solutions Programming Project 6.3: Rewrite the Sorts class so that both sorting algorithms put the values in descending order [CR3]
- Java Software Solutions Programming Project 6.6 (Savings Account): Design and implement a program that manages saving accounts. Keep track of key information and let each customer make deposits and withdrawals. Produce error messages for invalid transactions [CR4]
- Java Software Solutions Programing Project 6.8: Create a class DeckOfCards that stores 52 objects of the Card class written previously. Include methods to shuffle the deck, deal a card and report the number of cards left in the deck. Design and implement an application that uses this class [CR4]

- Design and implement an application that reads a sequence of up to 25 pairs of names and postal ZIP codes for individuals. Store the data in an object. Print the list in an appropriate format to the screen

Standards Addressed:

[CR2a]: The course teaches students to use and implement commonly used algorithms.

[CR2b]: The course teaches students to use commonly used data structures.

[CR3]: The course teaches students to select appropriate algorithms and data structures to solve problems.

[CR4]: The course teaches students to code fluently in an object-oriented paradigm using the programming language Java.

Unit #3: Semester 1/Week 7-9
Big Ideas: Inheritance
Text and Resources:
- Java Software Solutions for AP Computer Science 3rd Edition or later. Addison-Weslay, 2011
- Be Prepared for the AP Computer Science Exam in Java, 6th Edition, Andover, Mass.: Skylight Publishing, 2014.
- Java Software Solutions student files, teacher files, PowerPoint presentations, Test Package, additional resources
- CodingBat: http//codingbat.com/java

Content Summary:
- Derive new classes from existing ones
- How inheritance supports software reuse and extending a given class using inheritance
- Add and modify methods in child classes
- How to design and implement class hierarchies
- Polymorphism and how it can be done
- Visibility as it relates to inheritance
- Purpose and concept of method overriding
- Using class members
- Interfaces and abstract classes [CR5]
  - Why abstract classes are necessary
- Java library classes (Object) [CR5]
- Reading and understanding class specifications and relationships among classes ("is-a" and "has-a");
- Applying functional decomposition.

Assessments:
Formative
1. Clicker Questions – Java Software Solutions, Multiple Choice 7.1-10; True/False 7.1-10; AP-Style Multiple Choice 7.1-7.6.
2. Multiple Choice and Free Response Exam.

Summative
1. Java Software Solutions, sections 7.0-7.7; Summary of Key Concepts (redacted).
2. Elevens Activities 6-9 (APCS A Labs). [CR4]
3. Java Software Solutions, Self-Review Questions 7.1-7.12.

Sample Activity:
- Elevens Activities 6-9 – Board and AbstractBoard Classes (APCS A Labs). 21
- Java Software Solutions Programming Project 7.2: Make a class Employee with a name and salary. Make a class Manager inherit from Employee. Add an instance field, named department , of type String. Supply a method toString that prints the manager's name, department, and salary. Make a class Executive inherit from Manager. Supply appropriate toString methods

for all classes. Supply a test program that tests these classes and methods [CR4]
- Java Softtware Solutions Programming Project 7.1 MonetaryCoin Project that is derived from the Coin class presented before. Show that a monetary coin inherits its parents ability to be flipped [CR4]
- Java Software Solutions Programming Project 7.3: Design and implement a set of classes that define various types of reading material. Include data values that describe various attributes of the material. Include a main driver class to instantiate and exercise several of the classes [CR4]
- Load revised inheritance hierarchy program
- Produce an error in two ways by deleting the calls to super
- Fix the error in two ways: add a default constructor or make a call to the existing super constructor
- Use the private variable pages without changing it
- Try to change it to produce an error
- Fix the error by making it protected
- Try the toString Method with the Dictionary
- Override the toString Method for the Dictionary class
- The equals method. How does it work?
- Is the compareTo method inherited?
- Override the equals method for the book class: public boolean equals (Object o)
- Class Hierarchy, family tree
- Add a class to the book hierarchy that extends from dictionary which extends from book
- Add another class Novel that inherits from book . This class does not inherit from the dictionary class
- Parents and descendant relationships. Siblings relationships
- Show how it is not necessary to declare an instance of the book class
- Declare the class as abstract
- Show that abstract classes cannot be instantiated
- Consider the situation in which the implementation of a method differs completely from one child to another. Add an abstract method to the class (abstract public computeExpenses)
- Add the abstract modifier to the account class
- Demonstrate how abstract method are not needed
- Include an abstract method public void deductFees (prototype)
- Delete the abstract modifier to produce an error. Fix the error
- Add a class that inherits from the account class. Demonstrate how it is being forced to override abstract methods
- Fix error by overriding method or by declaring the class itself abstract
- Add another class that inherits from the account class
- Try to instantiate the abstract class to produce an error
- Can you instantiate an interface?
- Show how you can still have an object reference whose type is an abstract class
- Demonstrate polymorphic variables. The type of the object will be determine at run time (Savings account or Checking account?)

- Design a class hierarchy for the following : vehicle, boat, car, plane
- Consider where to include variable speed, method fuel consumption and variable number of wheels

Standards Addressed:

[CR4]: The course teaches students to code fluently in an object-oriented paradigm using the programming language Java.

[CR5]: The course teaches students to use elements of the standard Java library from the AP Java subset in Appendix A of the AP Computer Science A Course Description.

Unit #4: Semester 1/Weeks 10-12
Big Ideas: Recursion/Merge and Quick Sort
Text and Resources:
- Java Software Solutions for AP Computer Science 3rd Edition or later. Addison-Weslay, 2011
- Be Prepared for the AP Computer Science Exam in Java, 6th Edition, Andover, Mass.: Skylight Publishing, 2014.
- Java Software Solutions student files, teacher files, PowerPoint presentations, Test Package, additional resources
- CodingBat: http//codingbat.com/java

Content Summary:
- Explain the underlying ideas of recursion
- Examine recursive methods and processing steps
- Infinite recursion and discuss ways to avoid it
- Demonstrate the use of recursion to solve problems
- Use recursion in sorting (Merge Sort, Quick Sort) [CR2a]
- Compare the recursive sorting algorithms with other sorts [CR3]
- Explain when recursion should and should not be used [CR3]

Assessments:
Formative
1. Java Software Solutions, sections 8.0-8.3; Summary of Key Concepts (redacted).
2. Java Software Solutions, Self-Review Questions 8.1-8.9
3. Tracing Recursion Worksheet.
4. Java Software Solutions Short Answer Exercises

Summative
1. Clicker Questions – Java Software Solutions, Multiple Choice 8.1-8.10; True/False 8.1-8.10; AP-Style Multiple Choice 8.1-8.6.
2. Multiple Choice and Free Response Exam.

Sample Activity:
- Java Software Solutions Programming Project 8.1: Design and implement a recursive version of the PalindromeTester Program from chapter 3
- Java Software Solutions Programming Project 8.2: Design and implement a program for finding the greatest common factor by using the Euclid's algorithm. Implement the Euclid's algorithm recursively
- Write a recursive function that finds the index of the smallest element in the array
- Java Software Solutions Programming Project 8.7: Design and implement a recursive program that generates random Blurbs in a described alien program
- Java Software Solutions Programming Project 8.9: Design and implement a recursive program to determine and print the Nth line of the Pascal's Triangle.

- Type a method that returns the sum of the first n integers recursively
- Implement the binary search recursively
- Implement the selection sort recursively
- Implement the merge ssort and quick sort recursively
- Implement a search algorithm that returns true if a given element is found in the array
- Simplify above search by using a third parameter
- Implement a method that return the maximum element in array using recursion
- Solve the Towers of Hanoi recursively

Standards Addressed:

[CR1]: This course teaches students to design and implement computer based solutions to problems

[CR2a]: The course teaches students to use and implement commonly used algorithms.

[CR2b]: The course teaches students to use commonly used data structures.

[CR6]: The course includes a structured lab component comprised of a minimum of 20 hours of hands-on lab experiences.

[CR7]: The course teaches students to recognize the ethical and social implications of computer use.

Unit #5: Semester 1/Weeks 13-19
*Big Idea: Ethical and Social Implications of Computer Use, Enrichment
Activities*
Text and Resources:
- The College Board's AP Computer Science A Elevens Lab Student Guide
- The College Board's AP Computer Science A Magpie Chatbot Lab Student Guide
- The College Board's AP Computer Science A Picture Lab Student Guide
- Current media sources and Internet articles and blogs discussing ethical and social issues related to computer use.
- The College Board's Computer Science A Course Description.
- The College Board's Magpie, Picture Lab, and Elevens Labs Teacher Guides.

Content Summary:
- Responsible use of computer systems
- System reliability, privacy, intellectual property, legal issues
- Social and ethical ramifications of computer use   [CR7]

Assessments:
Formative
1. One student-chosen chapter of Blown to Bits.
2. Prepare a one-page summary of the chapter and participate in a classroom discussion of it. [CR7]
3. Java Software Solutions sections 1.5, 2.11, 2.12, 3.9, 4.6, 4.7, 5.6, 5.7, 6.8, 6.9, 7.8 and 8.4
4. One student-chosen chapter of Blown to Bits.
5. Prepare a one-page summary of the chapter and participate in a classroom discussion of it. [CR7]

Sample Activities:
- Computer graphics concepts
- The Java Graphics class
- GUI components and their events
- Layouts.
- Handling mouse and keyboard events and images.
- Text and binary files
- Streams vs. random-access files
- Java I/O package
- The Scanner class
- Checked exceptions

Standards Addressed:

[CR1]: This course teaches students to design and implement computer based solutions to problems

[CR2a]: The course teaches students to use and implement commonly used algorithms.

[CR2b]: The course teaches students to use commonly used data structures.

[CR6]: The course includes a structured lab component comprised of a minimum of 20 hours of hands-on lab experiences.

[CR7]: The course teaches students to recognize the ethical and social implications of computer use.