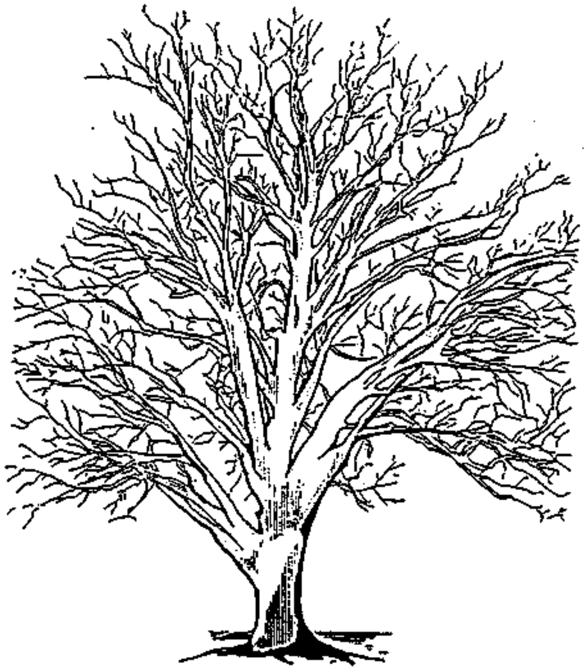
Monroe Township Schools



Curriculum Management System

Advanced Computer Science (H)

Grade 11-12

July 2008

*** For adoption by all regular education programs
as specified and for adoption or adaptation by
all Special Education Programs in accordance
with Board of Education Policy # 2220.**

Board Approved: August 2008

Table of Contents

MONROE TOWNSHIP SCHOOL DISTRICT

ADMINISTRATION

Mr. Joseph King, Interim Superintendent
Dr. Christopher H. Tienken, Assistant Superintendent
Dr. Veronica Geyer, Assistant Superintendent

BOARD OF EDUCATION

Ms. Amy Antelis, President
Mr. Lew Kaufman, Vice President
Mr. Marvin Braverman
Mr. Ken Chiarella
Ms. Kathy Kolupanowich
Mr. John Leary
Ms. Kathy Leonard
Mrs. Rita Ostrager
Mr. Ira Tessler

JAMESBURG REPRESENTATIVE

Ms. Patrice Faraone

Student Board Members

Ms. Melissa Bonamici
Ms. Nidhi Bhatt

Acknowledgments

The following individuals are acknowledged for their assistance in the preparation of this Curriculum Management System:

Writer's Name: **Nicole Santora, Teacher of Mathematics**

Supervisor Name: **Robert O'Donnell, Supervisor of Mathematics and Educational Technology**

Technology Staff: **Al Pulsinelli**
Reggie Washington
Bill Wetherill

Secretarial Staff: **Debby Gialanella**
Geri Manfre
Gail Nemeth

Monroe Township Schools
Mission and Goals

Mission

The mission of the Monroe Township School District, a unique multi-generational community, is to collaboratively develop and facilitate programs that pursue educational excellence and foster character, responsibility, and life-long learning in a safe, stimulating, and challenging environment to empower all individuals to become productive citizens of a dynamic, global society.

Goals

To have an environment that is conducive to learning for all individuals.

To have learning opportunities that are challenging and comprehensive in order to stimulate the intellectual, physical, social and emotional development of the learner.

To procure and manage a variety of resources to meet the needs of all learners.

To have inviting up-to-date, multifunctional facilities that both accommodate the community and are utilized to maximum potential.

To have a system of communication that will effectively connect all facets of the community with the Monroe Township School District.

To have a staff that is highly qualified, motivated, and stable and that is held accountable to deliver a safe, outstanding, and superior education to all individuals.

INTRODUCTION, PHILOSOPHY OF EDUCATION, AND EDUCATIONAL GOALS

Philosophy

Monroe Township Schools are committed to providing all students with a quality education resulting in life-long learners who can succeed in a global society. The mathematics program, grades K - 12, is predicated on that belief and is guided by the following six principles as stated by the National Council of Teachers of Mathematics (NCTM) in the *Principles and Standards for School Mathematics, 2000.* First, a mathematics education requires equity. All students will be given worthwhile opportunities and strong support to meet high mathematical expectations. Second, a coherent mathematics curriculum will effectively organize, integrate, and articulate important mathematical ideas across the grades. Third, effective mathematics teaching requires the following: a) knowing and understanding mathematics, students as learners, and pedagogical strategies b) having a challenging and supportive classroom environment and c) continually reflecting on and refining instructional practice. Fourth, students must learn mathematics with understanding. A student's prior experiences and knowledge will actively build new knowledge. Fifth, assessment should support the learning of important mathematics and provide useful information to both teachers and students. Lastly, technology enhances mathematics learning, supports effective mathematics teaching, and influences what mathematics is taught.

As students begin their mathematics education in Monroe Township, classroom instruction will reflect the best thinking of the day. Children will engage in a wide variety of learning activities designed to develop their ability to reason and solve complex problems. Calculators, computers, manipulatives, technology, and the Internet will be used as tools to enhance learning and assist in problem solving. Group work, projects, literature, and interdisciplinary activities will make mathematics more meaningful and aid understanding. Classroom instruction will be designed to meet the learning needs of all children and will reflect a variety of learning styles.

In this changing world those who have a good understanding of mathematics will have many opportunities and doors open to them throughout their lives. Mathematics is not for the select few but rather is for everyone. Monroe Township Schools are committed to providing all students with the opportunity and the support necessary to learn significant mathematics with depth and understanding. This curriculum guide is designed to be a resource for staff members and to provide guidance in the planning, delivery, and assessment of mathematics instruction.

Educational Goals

Advanced Computer Science (H) is the third level in the computer programming series. Students enrolled in this class have successfully completed a year of learning and programming using Visual Basic.Net in Computer Programming I and the Java programming language in AP Computer Science A. This class begins with an advanced study of the topics introduced in the AP Computer Science A course. The focus of this course is a more intense in depth study of algorithms, data structures, design, and abstraction. There is also an in depth study of the Grid World Case Study, where students will create their own classes to add to the study.

New Jersey State Department of Education
Core Curriculum Content Standards

A note about Mathematics Standards and Cumulative Progress Indicators.

The New Jersey Core Curriculum Content Standards for Mathematics were revised in 2008. The Cumulative Progress Indicators (CPI's) referenced in this curriculum guide refer to these new standards and may be found in the Curriculum folder on the district servers. A complete copy of the new Core Curriculum Content Standards for Mathematics may also be found at:

http://education.state.nj.us/cccs/?_standard_matrix;c=4

Advanced Computer Science (H)

Scope and Sequence

Quarter I	
Big Idea: Technology I. Fundamentals about Software a. define programming b. translate human-readable programs to machine code c. understand and recognize types of errors d. outline the compilation process	**Big Idea: Language** II. An Introduction to Objects a. understand variables b. understand the concepts of classes and objects c. call methods d. understand parameters and return values e. research different classes and methods using the Java API f. distinguish between an object and an object reference g. differentiate between the different levels of abstraction h. implement classes i. understand the purpose and use constructors j. access instance fields and local variables
Big Idea: Language III. Programming Fundamentals a. distinguish between number types b. recognize limitations of number types, overflow, and round off errors c. understand the proper use of constants d. perform arithmetic operations and use mathematical functions e. call static methods f. use String type to define and manipulate character strings g. read input and produce formatted output h. use the if statement i. compare primitive data and objects j. use for loops k. use while loops l. process sentinel values m. use random numbers and simulations	**Big Idea: Organization** IV. Arrays and Introduction to Algorithms a. create arrays b. create array lists c. utilize the wrapper classes and auto-boxing d. discuss the for each loop e. construct two dimensional arrays f. copy arrays

Quarter II	
Big Idea: Objects V. An Introduction to Class Design a. understand cohesion and coupling b. define accessors, mutators, and immutable classes c. comprehend side effects of a method d. discuss pre-conditions and post-conditions e. use static methods f. comprehend scope	**Big Idea: Objects** VI. Interfaces and Inheritance a. use interfaces b. convert between class and interface types c. understand concepts of polymorphism d. understand how interfaces can be used to decouple classes e. implement helper classes f. use inheritance g. create subclasses h. utilize polymorphism i. distinguish between access controls j. understand the affects of the Object class
Big Idea: Plan VII. Algorithms a. utilize recursion b. understand and reply relationship between recursion and iteration c. understand recursive helper methods d. discuss the efficiency of recursion e. analyze the algorithm and utilize the selection sort, insertion sort, quick sort, and the merge sort f. understand the big-Oh notation g. learn how to estimate and compare the performance of algorithms h. Utilize the binary search and the linear search	**Big Idea: Language** VIII. Files and Exceptions a. read and write text files b. throw exceptions c. differentiate between checked and unchecked exceptions d. catch exceptions e. design own exception classes

Quarter III	
Big Idea: Design IX. Object Oriented Design a. understand the software life cycle b. discover new classes and methods c. identify inheritance, aggregation, and dependency relationships between classes	**Big Idea: Efficiency** X. Introduction to Data Structures a. use the linked lists provided in the standard library b. use iterators to traverse linked lists c. understand the implementation of linked lists d. distinguish between abstract and concrete data types e. know the efficiency of fundamental operations of lists and arrays f. understand stack and queue types
Big Idea: Efficiency XI. Advanced Data Structures a. understand set and map data type b. understand the implementation of hash tables c. program hash tables d. employ binary trees e. use tree sets and tree maps f. comprehend heap data structures g. implement the priority queue data table h. understand how to use heaps for sorting	**Big Idea: Nature** XII. Grid World a. use and modify existing classes b. understand intercommunicating objects c. use Inheritance d. understand arrays e. use of abstract classes f. use of data structures
Quarter IV	
Big Idea: Legal XIII.Computing in Context a. comprehend system reliability b. be familiar with privacy issues c. understand legal issues and intellectual property d. be aware of social and ethical ramifications	**Big Idea: Numbers** XIV. Number Concepts a. convert decimal numbers to binary numbers b. understand and use the hexadecimal system
Big Idea: Business XV. Independent Assignment a. evaluate a pre-existing software used in business b. create an original, improved program that completes the same tasks as the existing program that was researched	

Suggested days of Instruction	**Curriculum Management System** **Grade Level/Subject:** **Advanced Computer Science (H)/ 11-12**	**Big Idea: Technology** **Topic: Fundamentals about Software** **Goal 1: The student will be able to understand how programming software works.**	
	Objectives / Cluster Concepts / Cumulative Progress Indicators (CPI's) **The student will be able to:**	**Essential Questions** **Sample Conceptual Understandings**	**Instructional Tools / Materials / Technology / Resources / Learning Activities / Interdisciplinary Activities / Assessment Model**
1	1.1 define programming (8.2.B.1)	• Explain the difference between using a computer program and programming a computer.	Instructional Tools/Materials/Technology/Resources • Cay Horstmann. *Java Concepts for AP Computer Science*. 5th ed. United States: John Wiley & Sons Inc., 2007. • Eclipse Programming Environment • Net Beans Programming Environment Learning Activities *Programs* • Create a flow chart demonstrating the compilation process- The compilation process involves many steps. Create a flow chart that shows your understanding of how the compilation process works. ○ Blooms Taxonomy – Knowledge, Comprehension, Application, Analysis, Synthesis *Students will use their understanding of the basics of Java to create the following programs:* • Face Printer – Create a program FacePrinter that prints a face using text characters. Use comments to indicate the statement that prints the hair, ears, mouth, and so on. ○ Blooms Taxonomy – Knowledge, Comprehension, Application
	1.2 translate human-readable programs to machine code (8.2.B.1)	• What is the Java Virtual Machine?	
	1.3 understand and recognize types of errors (4.5.B.1, 4.5.B.2, 4.5.B.3)	• What is the difference between syntax, logical, and run-time errors? How do you discover these errors?	
	1.4 outline the compilation process (8.2.B.1, 8.2.B.2)	• Explain in detail the compilation process.	

Suggested days of Instruction	**Curriculum Management System** **Grade Level/Subject:** **Advanced Computer Science (H)/ 11-12**	**Big Idea: Technology** **Topic: Fundamentals about Software** **Goal 1: The student will be able to understand how programming software works.**	
	Objectives / Cluster Concepts / Cumulative Progress Indicators (CPI's) **The student will be able to:**	**Essential Questions** **Sample Conceptual Understandings**	**Instructional Tools / Materials / Technology / Resources / Learning Activities / Interdisciplinary Activities / Assessment Model**
			• Sum of Reciprocals – Write a program Sum of Reciprocals that compputers the sum of the reciprocals 1/1 + ½ + … + 1/10. This is harder than it sounds. Try writing the program, and check the result. The program's result isn't likely to be correct. Then write the denominators as floating-point numbers, 1.0, 2.0, …, 10.0, and run the program again.. Can you explain the difference in the results? ○ Blooms Taxonomy – Knowledge, Comprehension, Application Assessment • Observation • Discussion • Quiz • Program Projects • Test

Suggested days of Instruction	**Curriculum Management System** **Grade Level/Subject:** **Advanced Computer Science (H)/ 11-12**	**Big Idea: Objects** **Topic: An Introduction to Objects** **Goal 2: The student will be able to write classes and create objects.**	
	Objectives / Cluster Concepts / Cumulative Progress Indicators (CPI's) **The student will be able to:**	**Essential Questions** **Sample Conceptual Understandings**	**Instructional Tools / Materials / Technology / Resources / Learning Activities / Interdisciplinary Activities / Assessment Model**
2	2.1 understand variables (4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2) 2.2 understand the concepts of classes and objects (4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2) 2.3 call methods (4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2) 2.4 understand parameters and return values (4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2) 2.5 research different classes and methods using the Java API (4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2) 2.6 distinguish between an object and an object reference (4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2)	• What are the rules that must be followed when creating a variable name? • Explain the difference between an object and a class. • Why is it necessary to put the object name when calling a method of a class? • What is the difference between actual and formal parameters? • Where is the Java API located? What does it list? • Explain the difference between an object and an object reference,	Instructional Tools/Materials/Technology/Resources • Cay Horstmann. *Java Concepts for AP Computer Science*. 5th ed. United States: John Wiley & Sons Inc., 2007. • Eclipse Programming Environment • Net Beans Programming Environment Learning Activities *Programs* *Students will use their understanding of the basics of Java to create the following programs:* • Area Tester – Write an AreaTester program that constructs a Rectangle object and then computes and prints its area. Use the getWidth and getHeight methods. Also print the expected answer. ○ Blooms Taxonomy – Knowledge, Comprehension, Application, Analysis, Synthesis • Lottery Printer – Write a program LotteryPrinter that picks a combination in a lottery. In this lottery, players can choose 6 numbers (possibly repeated) between 1 and 49. You program should print out a sentence such as “Play this combination – it’ll make you rich!”, followed by a lottery combination. ○ Blooms Taxonomy – Knowledge, Comprehension, Application,

Suggested days of Instruction	**Curriculum Management System** **Grade Level/Subject:** **Advanced Computer Science (H)/ 11-12**	**Big Idea: Objects** **Topic: An Introduction to Objects** **Goal 2: The student will be able to write classes and create objects.**	
	Objectives / Cluster Concepts / Cumulative Progress Indicators (CPI's) **The student will be able to:**	**Essential Questions** **Sample Conceptual Understandings**	**Instructional Tools / Materials / Technology / Resources / Learning Activities / Interdisciplinary Activities / Assessment Model**
	2.7 differentiate between the different levels of abstraction (4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2)	• What are the different levels of abstraction? How does a programmer know which ones to use?	Analysis, Synthesis • Gregorian Calendar – The GregorianCalendar class describes a point in time, as measured by the Gregorian calendar, the standard calendar that is commonly used throughout the world today. You construct a GregorianCalendar object from a year, month, and day of the month. Your task is to write a program that prints the following information: the date and weekday that is 100 days from today, the weekday of your birthday, the date that is 10,000 days from your birthday. ○ Blooms Taxonomy – Knowledge, Comprehension, Application, Analysis, Synthesis
	2.8 implement classes (4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2)	• How is a driver class created? What is the purpose of the driver class?	
	2.9 understand the purpose and use constructors (4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2)	• What is the purpose of a constructor? Does a programmer have to add one?	
	2.10access instance fields and local variables (4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2)	• Why are instance variables called instance variables and not just variables?	• Bank Account – Enhance the bank account class to see abstraction and encapsulation. Begin with a simple enhancement: charging a fee for every deposit and withdrawal. Supply a mechanism for setting the fee and modify the deposit and withdraw methods so that the fee is levied. Test your resulting class and check that the fee is computed correctly. Now make a more complex change. The bank will allow a fixed number of free transactions every month, and charge for transactions exceeding the free allotment. The charge is not levied immediately but at the end of the month. Supply a new method deductMonthlyCharge to the BankAccount class that deducts the monthly charge and resets the transaction

Suggested days of Instruction	Curriculum Management System Grade Level/Subject: **Advanced Computer Science (H)/ 11-12**	**Big Idea: Objects** **Topic: An Introduction to Objects** **Goal 2: The student will be able to write classes and create objects.**	
	Objectives / Cluster Concepts / Cumulative Progress Indicators (CPI's) **The student will be able to:**	**Essential Questions** **Sample Conceptual Understandings**	**Instructional Tools / Materials / Technology / Resources / Learning Activities / Interdisciplinary Activities / Assessment Model**
			count. Produce a test program that verifies that the fees are calculated correctly over several months. ○ Blooms Taxonomy – Knowledge, Comprehension, Application, Analysis, Synthesis • Greeter – Begin with a simple Greeter class that has single method, sayHello. That method should return a string, not print it. Enhance the Greeter class so that each object produces a customized greeting. Use the concat method of the String class. Add a method sayGoodBye and a method refuseHelp. ○ Blooms Taxonomy – Knowledge, Comprehension, Application Assessment • Observation • Discussion • Quiz • Program Projects • Test

Suggested days of Instruction	**Curriculum Management System** **Grade Level/Subject:** **Advanced Computer Science (H)/ 11-12**	**Big Idea: Language** **Topic: Programming Fundamentals** **Goal 3: The student will be able to effectively use primitive data, the String class, decision structures, and repetition structures.**	
	Objectives / Cluster Concepts / Cumulative Progress Indicators (CPI's) **The student will be able to:**	**Essential Questions** **Sample Conceptual Understandings**	**Instructional Tools / Materials / Technology / Resources / Learning Activities / Interdisciplinary Activities / Assessment Model**
4	3.1 distinguish between number types (4.1.B.1, 4.5.A.2, 4.5.C.2, 4.5.C.4, 4.5.F.2)	- How many different number types exist? Why is there more than one? Explain when you would use each?	Instructional Tools/Materials/Technology/Resources - Cay Horstmann. *Java Concepts for AP Computer Science*. 5th ed. United States: John Wiley & Sons Inc., 2007. - Eclipse Programming Environment - Net Beans Programming Environment
	3.2 recognize limitations of number types, overflow, and round off errors (4.1.C.1, 4.2.D.1, 4.2.D.2, 4.5.A.2, 4.5.C.2, 4.5.F.2)	- What is a numeric computation overflow? How can one be avoided? - Give an example of a floating-point round off error. Would the same example work correctly if used integers and switched to a sufficiently small unit, such as cents instead of dollars so that the values don't have a fractional part?	Learning Activities *Programs* *Students will use their understanding of the programming fundamentals to create the following programs:*
	3.3 understand the proper use of constants (4.5.A.2, 4.5.C.2, 4.5.F.2)	- What is a final variable? Can you define a final variable without supplying its value?	- Balloon – Design and implement a class Balloon that models a spherical balloon that is being filled with air. The constructor constructs an empty balloon. Supply these methods: addAir, getVolume, getSurfaceArea, getRadius. Supply a BalloonTester class that constructs a balloon, adds 100 cm^3 of air, tests the three accessor methods, add another 100 cm^3 of air, and tests the accessor methods again.
	3.4 perform arithmetic operations and use mathematical functions (4.1.B.1, 4.5.A.2, 4.5.C.2, 4.5.C.4, 4.5.F.2)	- Name two very important things to remember when rewriting a mathematical expression into java notation.	○ Blooms Taxonomy – Knowledge, Comprehension, Application, Analysis, Synthesis
	3.5 call static methods (4.5.A.2, 4.5.C.2, 4.5.F.2)	- Explain the difference between a static method and a regular method. Why is a programmer able to call a static method directly?	- Break Integer – Write a program that reads in an integer and breaks it into a sequence of individual digits in reverse order. You may assume that the input has no more than five digits and is not negative.

Suggested days of Instruction	**Curriculum Management System** **Grade Level/Subject:** **Advanced Computer Science (H)/ 11-12**	**Big Idea:** **Language** **Topic: Programming Fundamentals** **Goal 3:** **The student will be able to effectively use primitive data, the String class, decision structures, and repetition structures.**	
	Objectives / Cluster Concepts / Cumulative Progress Indicators (CPI's) **The student will be able to:**	**Essential Questions** **Sample Conceptual Understandings**	**Instructional Tools / Materials / Technology / Resources / Learning Activities / Interdisciplinary Activities / Assessment Model**
	3.6 use String type to define and manipulate character strings (4.5.A.2, 4.5.C.2, 4.5.F.2)	• Name two different ways to concatenate Strings. Which way do you prefer and why?	○ Blooms Taxonomy – Knowledge, Comprehension, Application, Analysis, Synthesis
	3.7 read input and produce formatted output (4.5.A.2, 4.5.C.2, 4.5.F.2)	• What class is used to read input and format output? What are the methods of this class?	• Military Time – Write a program that reads two times in military format and finds the interval between those times. Extra credit if you can deal with the case where the first time is later than the second time. Implement a class TimeInterval whose constructor takes two military times. The class should have two methods getHours and getMinutes. ○ Blooms Taxonomy – Knowledge, Comprehension, Application, Analysis, Synthesis
	3.8 use the if statement (4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2)	• Explain the difference between if/else if/ else statement and nested if statements. Give an example for each.	
	3.9 compare primitive data and objects (4.5.A.2, 4.5.C.2, 4.5.F.2)	• Explain the difference between “r == s” and “r.equals(s)” where both r and s are of type String.	• Rounding Error – Write a program that reads in two floating point numbers and tests whether they are the same when rounded to two decimal places and whether they differ by less than .01. ○ Blooms Taxonomy – Knowledge, Comprehension, Application, Analysis, Synthesis
	3.10 use for loops (4.2.B.4, 4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2)	• Give an example of a for loop in which symmetric bounds are more natural. Give an example of a for loop in which asymmetric bounds are more natural.	
	3.11 use while loops (4.2.B.4, 4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2)	• Give simple rules as to when to use a for loop and when to use a while loop.	• Conversion – Write a unit conversion program that asks users to identify the unit from which they want to convert and the unit to which they want to convert. Legal units are inches, feet, miles, millimeters, centimeters, meters, and kilometers. Define two objects of a class UnitConverter that convert between meters and a given unit.
	3.12 process sentinel values (4.2.B.4, 4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2)	• Sometimes students write programs with instructions such as “Enter data, 0 to quit?” and that exit the data entry loop when the user enters the number 0. Explain why this is usually a poor idea.	

Suggested days of Instruction	Curriculum Management System Grade Level/Subject: **Advanced Computer Science (H)/ 11-12**	**Big Idea:** Language **Topic: Programming Fundamentals** **Goal 3: The student will be able to effectively use primitive data, the String class, decision structures, and repetition structures.**	
	Objectives / Cluster Concepts / Cumulative Progress Indicators (CPI's) **The student will be able to:**	**Essential Questions** **Sample Conceptual Understandings**	**Instructional Tools / Materials / Technology / Resources / Learning Activities / Interdisciplinary Activities / Assessment Model**
	3.13 use random numbers and simulations (4.4.B.1, 4.4.B.3, 4.4.B.4, 4.5.A.2, 4.5.C.2, 4.5.F.2)	• How would use a random number generator to simulate the drawing of a playing card?	○ Blooms Taxonomy – Knowledge, Comprehension, Application, Analysis, Synthesis • Line program – Implement a class Line with four constructors corresponding to the four cases: two points, a point and a slope, an equation in slope intercept form, and an equation of a vertical line. The class should find the equation of the line and compare with another line to find if they are equal, parallel, or intersect. ○ Blooms Taxonomy – Knowledge, Comprehension, Application, Analysis, Synthesis • Projectile flight – Implement a class that utilizes the formulas to find accurate trajectories for objects. Have the user input the time and the velocity of the object. ○ Blooms Taxonomy – Knowledge, Comprehension, Application, Analysis, Synthesis • Fibonacci sequence – Write a program that prompts the user for n and prints the first n values in the Fibonacci. Use a class FibonacciGenerator with a method nextNumber. ○ Blooms Taxonomy – Knowledge, Comprehension, Application, Analysis, Synthesis

Suggested days of Instruction	**Curriculum Management System** **Grade Level/Subject:** **Advanced Computer Science (H)/ 11-12**	**Big Idea: Language** **Topic: Programming Fundamentals** **Goal 3: The student will be able to effectively use primitive data, the String class, decision structures, and repetition structures.**	
	Objectives / Cluster Concepts / Cumulative Progress Indicators (CPI's) **The student will be able to:**	**Essential Questions** **Sample Conceptual Understandings**	**Instructional Tools / Materials / Technology / Resources / Learning Activities / Interdisciplinary Activities / Assessment Model**
			• Heron method – Implement a class called RootApproximator that uses Heron's method for computing square roots of numbers. The class should start with an initial guess of 1 and whose nextGuess method produces a sequence of increasingly better guesses. Supply a method hasMoreGuesses that returns false if two successive guesses are sufficiently close to each other. ○ Blooms Taxonomy – Knowledge, Comprehension, Application, Analysis, Synthesis • Flesch Readability Index – Create a program that uses Flesch Readability Index to gauge the legibility of a document without linguistic analysis. The program should count all words in a paragraph, count all syllables in each word, count all sentences, and use the formula to compute the index. The index is a number between 0 to 100, that indicates how difficult the text is to read. ○ Blooms Taxonomy – Knowledge, Comprehension, Application, Analysis, Synthesis Assessment • Observation • Discussion • Quiz

Suggested days of Instruction	**Curriculum Management System** **Grade Level/Subject:** **Advanced Computer Science (H)/ 11-12**	**Big Idea: Language**	
		Topic: Programming Fundamentals	
		Goal 3: The student will be able to effectively use primitive data, the String class, decision structures, and repetition structures.	
	Objectives / Cluster Concepts / **Cumulative Progress Indicators (CPI's)** **The student will be able to:**	**Essential Questions** **Sample Conceptual Understandings**	**Instructional Tools / Materials / Technology / Resources / Learning Activities / Interdisciplinary Activities / Assessment Model**
			• Program Projects • Test

Suggested days of Instruction	**Curriculum Management System** **Grade Level/Subject:** **Advanced Computer Science (H)/ 11-12**	**Big Idea:** **Organization**	
		Topic: Arrays and Introduction to Algorithms	
		Goal 4: The student will be able to use array and array lists to simplify coding and understand basic algorithms.	
	Objectives / Cluster Concepts / Cumulative Progress Indicators (CPI's) **The student will be able to:**	**Essential Questions** **Sample Conceptual Understandings**	**Instructional Tools / Materials / Technology / Resources / Learning Activities / Interdisciplinary Activities / Assessment Model**
5	4.1 create arrays (4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2) 4.2 create array lists (4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2) 4.3 utilize the wrapper classes and auto-boxing (4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2) 4.4 discuss the for each loop (4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2) 4.5 construct two dimensional arrays (4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2) 4.6 copy arrays (4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2)	• What is an out of bounds error? How can one be avoided? • Compare and contrast arrays and array lists. Give examples when it would be best to use each. • Why would a programmer need to wrap a variable? What are the advantages of using auto boxing? • What can a for each loop accomplish? What are its limitations? • Can two dimensional arrays have different data types? Can they be returned by a method? • When an array is copied, is an alias created? Explain the process of copying arrays.	Instructional Tools/Materials/Technology/Resources • Cay Horstmann. *Java Concepts for AP Computer Science*. 5th ed. United States: John Wiley & Sons Inc., 2007. • Eclipse Programming Environment • Net Beans Programming Environment Learning Activities *Programs* *Students will use their understanding of arrays and algorithms to create the following programs:* • Random Permutations – Write a program that produces random permutations of the numbers 1 to 10. To generate a random permutation, you need to fill an array with the numbers 1 to 10 so that no two entries of the array have the same contents. You could do it by brute force by calling Random.nextInt until it produces a value that is not yet in the array. Instead you should implement a smart method. Make a second array and fill it with the numbers 1 to 10. Then pick one of those at random, remove it, and append it to the permutation array. Repeat 10 times. Implement a class PermutationGenerator with a method nextPermutation. ○ Blooms Taxonomy – Knowledge, Comprehension, Application, Analysis, Synthesis

Suggested days of Instruction	Curriculum Management System Grade Level/Subject: **Advanced Computer Science (H)/ 11-12**	**Big Idea:** Organization **Topic: Arrays and Introduction to Algorithms** **Goal 4: The student will be able to use array and array lists to simplify coding and understand basic algorithms.**	
	Objectives / Cluster Concepts / Cumulative Progress Indicators (CPI's) **The student will be able to:**	**Essential Questions** **Sample Conceptual Understandings**	**Instructional Tools / Materials / Technology / Resources / Learning Activities / Interdisciplinary Activities / Assessment Model**
			• Magic Squares – Write a program that reads in n^2 values from the keyboard and tests whether they form a magic square when arranged as a square matrix. A magic square is an n x n matrix that is filled with the numbers 1, 2, 3,…,n^2. The sum of the elements of each row, in each column , and in the two diagonals is the same value. ○ Blooms Taxonomy – Knowledge, Comprehension, Application, Analysis, Synthesis • Poker Simulation – Create a program that implements a simulation of video poker. The card deck of 52 cards with 13 of each suit. At the beginning of the game the deck is shuffled. You need to devise a fair method for shuffling. The top five cards of the deck are presented to the player. The player can reject none, some or all of the cards. The rejected cards are replaced from the top of the deck. Use basic poker rules to score the deck. ○ Blooms Taxonomy – Knowledge, Comprehension, Application, Analysis, Synthesis Assessment • Observation • Discussion • Quiz

<table>
<tr><td rowspan="4">Suggested days of Instruction</td><td rowspan="3"><b>Curriculum Management System</b>
<b><u>Grade Level/Subject</u>:</b>
<b>Advanced Computer Science (H)/ 11-12</b></td><td colspan="2"><b>Big Idea: Organization</b></td></tr>
<tr><td colspan="2"><b>Topic: Arrays and Introduction to Algorithms</b></td></tr>
<tr><td colspan="2"><b><u>Goal 4:</u> The student will be able to use array and array lists to simplify coding and understand basic algorithms.</b></td></tr>
<tr><td><b>Objectives / Cluster Concepts /</b>
<b>Cumulative Progress Indicators (CPI's)</b>
<b>The student will be able to:</b></td><td><b>Essential Questions</b>
<b>Sample Conceptual Understandings</b></td><td><b>Instructional Tools / Materials / Technology / Resources / Learning Activities / Interdisciplinary Activities / Assessment Model</b></td></tr>
<tr><td></td><td></td><td></td><td>• Program Projects
• Test</td></tr>
</table>

Suggested days of Instruction	**Curriculum Management System** **Grade Level/Subject:** **Advanced Computer Science (H)/ 11-12**	**Big Idea: Objects**	
		Topic: Introduction to Class Design	
		Goal 5: The student will be able to design efficient classes.	
	Objectives / Cluster Concepts / Cumulative Progress Indicators (CPI's) **The student will be able to:**	**Essential Questions** **Sample Conceptual Understandings**	**Instructional Tools / Materials / Technology / Resources / Learning Activities / Interdisciplinary Activities / Assessment Model**
5	5.1 understand cohesion and coupling (4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2)	• What is coupling? Why is it a good idea to minimize coupling between classes?	Instructional Tools/Materials/Technology/Resources • Cay Horstmann. *Java Concepts for AP Computer Science*. 5th ed. United States: John Wiley & Sons Inc., 2007. • Eclipse Programming Environment • Net Beans Programming Environment
	5.2 define accessors, mutators, and immutable classes (4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2)	• What is the difference between an accessor method and a mutator method? • What is an immutable class?	
	5.3 comprehend side effects of a method (4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2)	• Ideally a method should have no side effects. Can you write a program in which no method has a side effect? Would such a program be useful?	Learning Activities *Programs* *Students will use their understanding of the class design to create the following programs:*
	5.4 discuss pre-conditions and post-conditions (4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2)	• When a method is called with parameters that violate its pre-condition, it can terminate (by throwing an exception or an assertion error), or it can return to its caller. Give two examples of library methods that return some result to their callers when called with invalid parameters and five two examples of library methods that terminate.	• Exponents – Consider the following algorithm for computing x^n for an integer n. If n<0, x^n is $1/ x^{-n}$. If n is positive and even, then $x^n = (x^{n/2})^2$. If n is positive and odd then $x^{n =} x^{n-1}(x)$. Implement a static method double intPower(double x, int n) that uses this algorithm. Add it to a class called Numeric. ○ Blooms Taxonomy – Knowledge, Comprehension, Application, Analysis, Synthesis
	5.5 use static methods (4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2)	• Give an example of when using a static method would be advantageous.	• Geometry – Create a program that has static methods that compute the volume and surface area of a sphere with radius r, a cylinder with a circular base with radius r and height h, and a cone with a circular base with radius r and height h. Place them into a class Geometry. Then write a program hat prompts the user for the values of r and h, calls the six methods,
	5.6 comprehend scope (4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2)	• What is the scope of a variable? Why does scope need to be regulated?	

Suggested days of Instruction	**Curriculum Management System** **Grade Level/Subject:** **Advanced Computer Science (H)/ 11-12**	**Big Idea: Objects** **Topic: Introduction to Class Design** **Goal 5: The student will be able to design efficient classes.**	
	Objectives / Cluster Concepts / Cumulative Progress Indicators (CPI's) **The student will be able to:**	**Essential Questions** **Sample Conceptual Understandings**	**Instructional Tools / Materials / Technology / Resources / Learning Activities / Interdisciplinary Activities / Assessment Model**
			and prints the results. ○ Blooms Taxonomy – Knowledge, Comprehension, Application, Analysis, Synthesis • Zip code – Using the formula created by the Unites States Postal Service, to implement a program that asks the user for a zip code and prints the bar code. Your program should also be able to carry out the opposite conversion. ○ Blooms Taxonomy – Knowledge, Comprehension, Application, Analysis, Synthesis Assessment • Observation • Discussion • Quiz • Program Projects • Test

Suggested days of Instruction	**Curriculum Management System** **Grade Level/Subject:** **Advanced Computer Science (H)/ 11-12**	**Big Idea: Objects** **Topic: Interfaces and Inheritance** **Goal 6: The student will be able to expand and create a hierarchy of classes.**	
	Objectives / Cluster Concepts / Cumulative Progress Indicators (CPI's) **The student will be able to:**	**Essential Questions** **Sample Conceptual Understandings**	**Instructional Tools / Materials / Technology / Resources / Learning Activities / Interdisciplinary Activities / Assessment Model**
5	6.1 Interfaces (4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2)	• Why do interface types make code more reusable? • How are interfaces useful when multiple programmers are working on a project?	Instructional Tools/Materials/Technology/Resources • Cay Horstmann. *Java Concepts for AP Computer Science*. 5th ed. United States: John Wiley & Sons Inc., 2007. • Eclipse Programming Environment • Net Beans Programming Environment
	6.2 convert between class and interface types (4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2)	• How does a cast such as (BankAccount) x differ from a cast of number values such as (int) x?	Learning Activities *Programs* *Students will use their understanding of interfaces and inheritance to create the following programs:*
	6.3 understand concepts of polymorphism (4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2)	• In Java a method call such as x.f() uses late binding- the exact method to be called depends on the type of the object to which x refers. Give tow kinds of method calls that use early binding in Java.	• Game Boards – Design a general program for managing board games with two players. Your program should be flexible enough to handle games such as tic-tac-toe or chess. Design an interface Game that describes a board game. Think about what your program needs to do. It asks the first player to input a move- a string in a game specific format such as B3 in chess. Your program knows nothing about specific games, so the Game interface must have a method to check if the move is valid. Once the move is found valid, it needs to be executed. Next the program needs to check whether the game is over. Supply two game programs that differ only in the initialization of the Game reference. ○ Blooms Taxonomy – Knowledge, Comprehension, Application, Analysis, Synthesis
	6.4 understand how interfaces can be used to decouple classes (4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2)	• How can using interfaces be used to decouple classes?	
	6.5 implement helper classes (4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2)	• What is a helper class? Create an example of one.	
	6.6 use inheritance	• Create an example of when a programmer may use	

Suggested days of Instruction	Curriculum Management System Grade Level/Subject: **Advanced Computer Science (H)/ 11-12**	**Big Idea: Objects** **Topic: Interfaces and Inheritance** **Goal 6: The student will be able to expand and create a hierarchy of classes.**	
	Objectives / Cluster Concepts / Cumulative Progress Indicators (CPI's) **The student will be able to:**	**Essential Questions** **Sample Conceptual Understandings**	**Instructional Tools / Materials / Technology / Resources / Learning Activities / Interdisciplinary Activities / Assessment Model**
	(4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2)	inheritance. What are the advantages to inheritance?	
	6.7 create subclasses (4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2)	• If you had a class Bank Account, what would the possible subclasses be? What would be the methods of the super class? What would be the methods of the subclasses?	• Work Force – Make a class Employee with a name and salary. Make a class Manager that inherits 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 that inherits from Manager. Supply appropriate toString methods for all classes. Supply a test program that tests these classes and methods. ○ Blooms Taxonomy – Knowledge, Comprehension, Application, Analysis, Synthesis
	6.8 utilize polymorphism (4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2)	• What is method overriding? Does a polymorphic object call the method that was overrode in the parent class or the one in the child class?	
	6.9 distinguish between access controls (4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2)	• Give an example of when a programmer would want a super class to have private instance data.	• Robot Maze – Create a program with robots with varying behavior. The robots try to escape a maze. A robot has a position and a method void move (Maze m) that modifies the position. Provide a common superclass Robot whose move method does nothing. Provide subclasses RandomRobot, RightHandRuleRobot, and MemoryRobot. Each of these robots has a different strategy for escaping. ○ Blooms Taxonomy – Knowledge, Comprehension, Application, Analysis, Synthesis
	6.10understand the affects of the Object class (4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2)	• What is the object class? What methods does it have? Can the object class be accessed directly?	Assessment • Observation

Suggested days of Instruction	**Curriculum Management System** **Grade Level/Subject:** **Advanced Computer Science (H)/ 11-12**	**Big Idea: Objects**	
		Topic: Interfaces and Inheritance	
		Goal 6: The student will be able to expand and create a hierarchy of classes.	
	Objectives / Cluster Concepts / Cumulative Progress Indicators (CPI's) **The student will be able to:**	**Essential Questions** **Sample Conceptual Understandings**	**Instructional Tools / Materials / Technology / Resources / Learning Activities / Interdisciplinary Activities / Assessment Model**
			• Discussion • Quiz • Program Projects • Test

Suggested days of Instruction	**Curriculum Management System** **Grade Level/Subject:** **Advanced Computer Science (H)/ 11-12**	**Big Idea: Plan** **Topic: Algorithms** **Goal 7: The student will be able to understand and use the recursive process and analyze pre-existing algorithms.**	
	Objectives / Cluster Concepts / Cumulative Progress Indicators (CPI's) **The student will be able to:**	**Essential Questions** **Sample Conceptual Understandings**	**Instructional Tools / Materials / Technology / Resources / Learning Activities / Interdisciplinary Activities / Assessment Model**
8	7.1 utilize recursion (4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2) 7.2 understand and reply relationship between recursion and iteration (4.4.C.4, 4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2) 7.3 understand recursive helper methods (4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2) 7.4 discuss the efficiency of recursion (4.4.C.4, 4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2) 7.5 analyze the algorithm and utilize the selection sort, insertion sort, quick sort, and the merge sort (4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2)	• Explain in your own words the recursive process. • Differentiate between recursion and iteration. When is more advantageous to use recursion? Iteration? • What is a recursive helper method? What purpose does it serve? • Give three examples when using recursion would be beneficial to solving a problem. • Compare and contrast the selection sort, insertion sort, quick sort, and merge sort. When would it be more advantageous to use each of these?	Instructional Tools/Materials/Technology/Resources • Cay Horstmann. *Java Concepts for AP Computer Science*. 5th ed. United States: John Wiley & Sons Inc., 2007. • Eclipse Programming Environment • Net Beans Programming Environment • Towers of Hanoi Internet Game Learning Activities *Programs* *Students will use their understanding of the recursion to create the following programs:* • Find It – Create a program that uses recursion to implement a method that tests whether a string is contained in a sentence. ○ Blooms Taxonomy – Knowledge, Comprehension, Application, Analysis, Synthesis • Towers of Hanoi – Create the Towers of Hanoi puzzle. It consists of N disks is solved by moving N-1 disks out of the way onto an extra peg, moving the largest disk to the last peg, then mvoing the N-1 disks from the extra peg to the last peg. This solution is recursive because, to move the substack of N-1 disks, we can use the same process. ○ Blooms Taxonomy – Knowledge, Comprehension, Application

Suggested days of Instruction	**Curriculum Management System** **Grade Level/Subject:** **Advanced Computer Science (H)/ 11-12**	**Big Idea: Plan** **Topic: Algorithms** **Goal 7: The student will be able to understand and use the recursive process and analyze pre-existing algorithms.**	
	Objectives / Cluster Concepts / Cumulative Progress Indicators (CPI's) **The student will be able to:**	**Essential Questions** **Sample Conceptual Understandings**	**Instructional Tools / Materials / Technology / Resources / Learning Activities / Interdisciplinary Activities / Assessment Model**
	7.6 understand the big-Oh notation (4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2) 7.7 learn how to estimate and compare the performance of algorithms (4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2) 7.8 utilize the binary search and the linear search (4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2)	• What is the growth rate of a standard algorithm to find the minimum value of an array? Of finding both the minimum and the maximum? • Suppose algorithm A takes 5 seconds to handle a data set of 1,000 records. If the algorithm A is an *O(n)* algorithm, how long will it take to handle a data set of 2,000 records? Of 10,000 records? • Why does the insertion sort perform significantly better than the selection sort if an array is already sorted? • How does the linear search find an element in an array? • How does the binary search find an element in an array? • When is it beneficial to use the binary search over the linear search?	• Escaping a Maze – Create a program to escape a maze. Use the following recursive approach to check whether you can escape from the maze. If you are at an exit, return true. Recursively check whether you can escape from one of the empty neighboring locations without visiting the current location. This method merely tests whether there is a path out of the maze. Extra credit if you can print out a path that leads to an exit. ○ Blooms Taxonomy – Knowledge, Comprehension, Application, Analysis, Synthesis • Fibonacci – Write a program that prompts the user for n and prints the first n values in the Fibonacci. Use recursion instead of iteration. In order to speed up the process keep track of the values that have already been completed. Compare the running time of your recursion implementation with that of the iterative implementation. ○ Blooms Taxonomy – Knowledge, Comprehension, Application, Analysis, Synthesis • Selection Sort – Write a program that generates a table of sample runs of the selection sort times automatically. The program should ask for the smallest and the largest value of n and the number of measurements and then make

Suggested days of Instruction	**Curriculum Management System** **Grade Level/Subject:** **Advanced Computer Science (H)/ 11-12**	**Big Idea: Plan** **Topic: Algorithms** **Goal 7: The student will be able to understand and use the recursive process and analyze pre-existing algorithms.**	
	Objectives / Cluster Concepts / Cumulative Progress Indicators (CPI's) **The student will be able to:**	**Essential Questions** **Sample Conceptual Understandings**	**Instructional Tools / Materials / Technology / Resources / Learning Activities / Interdisciplinary Activities / Assessment Model**
			all sample runs. ○ Blooms Taxonomy – Knowledge, Comprehension, Application, Analysis, Synthesis • Merge Sort – Implement the sort method of the merge sort algorithm without recursion, where the length of the array is a power of two. First merge adjacent regions of size 1, then adjacent regions of size 2, then adjacent regions of size 4, and so on. ○ Blooms Taxonomy – Knowledge, Comprehension, Application, Analysis, Synthesis • Sort and Search – Create a program that uses the insertion sort and the binary search to search an array of any kind. Implement this algorithm and measure its performance. ○ Blooms Taxonomy – Knowledge, Comprehension, Application, Analysis, Synthesis • Person – Create a class Person that implements the Comparable interface. Compare persons by their name. Ask the user to input 10 names and generate 10 Person objects. Using the compareTo method, determine the first and last person among them and print them. ○ Blooms Taxonomy – Knowledge, Comprehension, Application,

Suggested days of Instruction	**Curriculum Management System** **Grade Level/Subject:** **Advanced Computer Science (H)/ 11-12**	**Big Idea: Plan**	
		Topic: Algorithms	
		Goal 7: The student will be able to understand and use the recursive process and analyze pre-existing algorithms.	
	Objectives / Cluster Concepts / Cumulative Progress Indicators (CPI's) **The student will be able to:**	**Essential Questions** **Sample Conceptual Understandings**	**Instructional Tools / Materials / Technology / Resources / Learning Activities / Interdisciplinary Activities / Assessment Model**
			Analysis, Synthesis • Appointment Book – Write a program that keeps an appointment book. Make a class Appointment that stores a description of the appointment, the appointment day, starting time, and ending time. Your program should keep the appointments in a sorted array list. Users can add appointments and print out all appointments for a given day. When a new appointment is added, use binary search to find where it should be inserted. Do not add if it conflicts with other appointments. ○ Blooms Taxonomy – Knowledge, Comprehension, Application, Analysis, Synthesis Assessment • Observation • Discussion • Quiz • Program Projects • Test

Suggested days of Instruction	**Curriculum Management System** **Grade Level/Subject:** **Advanced Computer Science (H)/ 11-12**	**Big Idea:** **Language** **Topic: Files and Exceptions** **Goal 8: The student will be able to input and output to a file and handle exceptions.**	
	Objectives / Cluster Concepts / Cumulative Progress Indicators (CPI's) **The student will be able to:**	**Essential Questions** **Sample Conceptual Understandings**	**Instructional Tools / Materials / Technology / Resources / Learning Activities / Interdisciplinary Activities / Assessment Model**
	8.1 read and write text files (4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2) 8.2 throw exceptions (4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2) 8.3 differentiate between checked and unchecked exceptions (4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2) 8.4 catch exceptions (4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2) 8.5 design own exception classes (4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2)	• What happens when you try to open a file for reading that doesn't exist? What happens if you try to open a file for writing that doesn't exist? • What is the difference between throwing an exception and catching an exception? • What is a checked exception? What is an unchecked exception? Is a NullPointerException checked or unchecked? Which exceptions do you need to declare with the throws keyword? • What happens if an exception does not have a matching catch clause? • What can your program do with the exception object that a catch clause receives? • Design your own exception class. Make sure you include all necessary parts.	Instructional Tools/Materials/Technology/Resources • Cay Horstmann. *Java Concepts for AP Computer Science*. 5th ed. United States: John Wiley & Sons Inc., 2007. • Eclipse Programming Environment • Net Beans Programming Environment Learning Activities *Programs* *Students will use their understanding of files and exceptions to create the following programs:* • File Info – Write a program that asks a user for a file name and prints the number of characters, words, and lines in that file. ○ Blooms Taxonomy – Knowledge, Comprehension, Application, Analysis, Synthesis • Concatenate – Write a program that concatenates the contents of several files into one file. The output file is always the last file specified on the command line. ○ Blooms Taxonomy – Knowledge, Comprehension, Application, Analysis, Synthesis • Reverse – Write a program that replaces each line of a file with its reverse. Of course, if you run Reverse twice on the same file, you get

Suggested days of Instruction	Curriculum Management System Grade Level/Subject: **Advanced Computer Science (H)/ 11-12**	**Big Idea:** Language **Topic: Files and Exceptions** **Goal 8: The student will be able to input and output to a file and handle exceptions.**	
	Objectives / Cluster Concepts / Cumulative Progress Indicators (CPI's) The student will be able to:	**Essential Questions Sample Conceptual Understandings**	**Instructional Tools / Materials / Technology / Resources / Learning Activities / Interdisciplinary Activities / Assessment Model**
			back the original file. ○ Blooms Taxonomy – Knowledge, Comprehension, Application, Analysis, Synthesis • Floating Error – Write a program that asks the user to enter floating point values. When the user enters a value that is not a number, give the user a second chance to enter the value. After two chances, quit reading input. Add all correctly specified values and print the sum when the user is done entering data. Use exception handling to detect improper inputs. ○ Blooms Taxonomy – Knowledge, Comprehension, Application, Analysis, Synthesis • Webpage – read the contents of a webpage and throw an exception if you find a malformed hyperlink. Extra credit if your program can follow the links that it finds and find links in those web pages as well. (This is the method search engines such as Google use to find web sites.) ○ Blooms Taxonomy – Knowledge, Comprehension, Application, Analysis, Synthesis Assessment • Observation

Suggested days of Instruction	**Curriculum Management System** **Grade Level/Subject:** **Advanced Computer Science (H)/ 11-12**	**Big Idea: Language**	
		Topic: Files and Exceptions	
		Goal 8: The student will be able to input and output to a file and handle exceptions.	
	Objectives / Cluster Concepts / Cumulative Progress Indicators (CPI's) **The student will be able to:**	**Essential Questions** **Sample Conceptual Understandings**	**Instructional Tools / Materials / Technology / Resources / Learning Activities / Interdisciplinary Activities / Assessment Model**
			• Discussion • Quiz • Program Projects • Test

Suggested days of Instruction	**Curriculum Management System** **Grade Level/Subject:** **Advanced Computer Science (H)/ 11-12**	**Big Idea: Design**	
		Topic: Object Oriented Design	
		Goal 9: The student will be able to comprehend object oriented programming and analyze the relationships formed between classes.	
	Objectives / Cluster Concepts / Cumulative Progress Indicators (CPI's) **The student will be able to:**	**Essential Questions** **Sample Conceptual Understandings**	**Instructional Tools / Materials / Technology / Resources / Learning Activities / Interdisciplinary Activities / Assessment Model**
2	9.1 understand the software life cycle (4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2, 8.2.B.11) 9.2 discover new classes and methods (4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2, 8.2.B.11) 9.3 identify inheritance, aggregation, and dependency relationships between classes (4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2, 8.2.B.11)	• What is the software life cycle? • List the steps in the process of object-oriented design that this chapter recommends for student use. • Give a rule of thumb for how to find classes when designing a program. • Give a rule of thumb for how to find methods when designing a program. • After discovering a method, why is it important to identify the object that is responsible for carrying out the action? • Every BMW is a vehicle. Should a class BMW inherit for the class vehicle? BMW is a vehicle manufacturer. Does that mean that the class BMW should inherit from the class VehicleManufacturer? • Explain aggregation in your own words.	Instructional Tools/Materials/Technology/Resources • Cay Horstmann. *Java Concepts for AP Computer Science*. 5th ed. United States: John Wiley & Sons Inc., 2007. • Eclipse Programming Environment • Net Beans Programming Environment Learning Activities *Programs* *Students will use their understanding of the object oriented design to create the following programs:* • Arithmetic Program – Write a program that teaches arithmetic to young children. The program tests addition and subtraction. In level 1 it tests only addition of numbers less than 10 whose sum is less than 10. In level 2 it tests addition of arbitrary one digit numbers. In level 3 it tests subtraction of one-digit numbers with a non negative difference. Generate random problems and get the player input. The player gets up to two tries per problem. Advance form one level to the next when the player has achieved a score of five points. ○ Blooms Taxonomy – Knowledge, Comprehension, Application, Analysis, Synthesis • Vending Machine – Write a program that simulates a vending machine. Products can be purchased by inserting coins with a value at

Suggested days of Instruction	**Curriculum Management System** **Grade Level/Subject:** **Advanced Computer Science (H)/ 11-12**	**Big Idea: Design** **Topic: Object Oriented Design** **Goal 9: The student will be able to comprehend object oriented programming and analyze the relationships formed between classes.**	
	Objectives / Cluster Concepts / Cumulative Progress Indicators (CPI's) **The student will be able to:**	**Essential Questions** **Sample Conceptual Understandings**	**Instructional Tools / Materials / Technology / Resources / Learning Activities / Interdisciplinary Activities / Assessment Model**
			least equal to the cost of the product. A user selects a product from a list of available products, add coins, and either gets the product or get the coins returned if insufficient money was supplied or if the product is sold out. The machine does not give change if too much money was added. Products can be restocked and money removed by an operator. Your solution should include a class VendingMachine that is not coupled with the Scanner or PrintStream classes. o Blooms Taxonomy – Knowledge, Comprehension, Application, Analysis, Synthesis • Airline Seating – Write a program that assigns seats on an airplane. Assume the airplane has 20 seats in first class and 90 seats in economy class. Your program should take three commands: add passengers, show seating, and quit. When passengers are added, ask for the class, the number of passengers traveling together, and the seating preference. Try to find a match and assign the seats. If no match exists, print a message. o Blooms Taxonomy – Knowledge, Comprehension, Application, Analysis, Synthesis • Tester – Write a program that administers and grades quizzes. A quiz consists of questions. There are four types of questions: text questions, number questions, choice questions

Suggested days of Instruction	**Curriculum Management System** **Grade Level/Subject:** **Advanced Computer Science (H)/ 11-12**	**Big Idea: Design**	
		Topic: Object Oriented Design	
		Goal 9: The student will be able to comprehend object oriented programming and analyze the relationships formed between classes.	
	Objectives / Cluster Concepts / Cumulative Progress Indicators (CPI's) **The student will be able to:**	**Essential Questions** **Sample Conceptual Understandings**	**Instructional Tools / Materials / Technology / Resources / Learning Activities / Interdisciplinary Activities / Assessment Model**
			with multiple answers. When grading a text question, ignore leading or trailing spaces and letter case. When grading a numeric question, accept a response that is approximately the same answer. A quiz is specified in a text file. Each question starts with a letter indicating the question type, followed by a line containing the text question. Your program should read in a quiz file, prompt the user for responses to all questions, and grade the responses. o Blooms Taxonomy – Knowledge, Comprehension, Application, Analysis, Synthesis Assessment • Observation • Discussion • Quiz • Program Projects • Test

Suggested days of Instruction	**Curriculum Management System** **Grade Level/Subject:** **Advanced Computer Science (H)/ 11-12**	**Big Idea: Efficiency** **Topic: Introduction to Data Structures** **Goal 10: The student will be able to understand and utilize a linked list and distinguish between concrete and abstract data types.**	
	Objectives / Cluster Concepts / Cumulative Progress Indicators (CPI's) **The student will be able to:**	**Essential Questions** **Sample Conceptual Understandings**	**Instructional Tools / Materials / Technology / Resources / Learning Activities / Interdisciplinary Activities / Assessment Model**
12	10.1 use the linked lists provided in the standard library (4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2)	• The linked list class in the Java library supports operations addLast and removeLast. To carry out these operations efficiently, the LinkedLIst class has an added reference last to the last node in the linked list. Draw a "before/after" diagram of the changes of the links in a linked list under the addLast and removeLast methods.	Instructional Tools/Materials/Technology/Resources • Cay Horstmann. *Java Concepts for AP Computer Science*. 5th ed. United States: John Wiley & Sons Inc., 2007. • Eclipse Programming Environment • Net Beans Programming Environment Learning Activities *Programs* *Students will use their understanding of data structures to create the following programs:*
	10.2 use iterators to traverse linked lists (4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2)	• The linked list class in the Java library supports bidirectional iterators. To go backward efficiently, each Node has an added reference, previous, to the predecessor node in the linked list. Draw a "before/after" diagram of the changes of the links in a linked list under the addFirst and removeFirst methods that shows how the previous links need to be updated.	• Reverse – Add a method reverse to the implementation of the LinkedList class that reverses the links in the list. Implement this method by directly rerouting the links, not by using an iterator. ○ Blooms Taxonomy – Knowledge, Comprehension, Application, Analysis, Synthesis
	10.3 understand the implementation of linked lists (4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2)	• Explain in your own words the implementation of linked lists.	
	10.4 distinguish between abstract and concrete data types (4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2)	• What is the difference between abstract and concrete data types? Give an example of when you would want an abstract data type? A concrete data type?	• Size - Add a method size to the implementation of the LinkedList class that computes the number of elements in the list, by following links and counting the elements until the end of the list is reached. ○ Blooms Taxonomy – Knowledge, Comprehension, Application, Analysis, Synthesis
	10.5 know the efficiency of	• What advantages do lists have over arrays? What	• Stack – The standard Java library implements a

Suggested days of Instruction	**Curriculum Management System** **Grade Level/Subject:** **Advanced Computer Science (H)/ 11-12**	**Big Idea:** **Efficiency** **Topic: Introduction to Data Structures** **Goal 10: The student will be able to understand and utilize a linked list and distinguish between concrete and abstract data types.**	
	Objectives / Cluster Concepts / Cumulative Progress Indicators (CPI's) **The student will be able to:**	**Essential Questions** **Sample Conceptual Understandings**	**Instructional Tools / Materials / Technology / Resources / Learning Activities / Interdisciplinary Activities / Assessment Model**
	fundamental operations of lists and arrays (4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2)	disadvantages do they have? • Suppose you needed to keep a collection of appointments. Would you use a list or an array of Appointment objects?	stack class, but in this exercise you are asked to provide your own implementation. Do not implement type parameters. Use an Object [] array to hold the stack elements. When the array fills up, allocate an array of twice the size and copy the values to the larger array. ○ Blooms Taxonomy – Knowledge, Comprehension, Application, Analysis, Synthesis
	10.6 understand stack and queue types (4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2)	• Suppose you write a program that models a card deck. Cards are taken from the top of the deck and given out to players. As cards are returned to the deck, they are placed on the bottom of the deck. Would you store the cards in a stack or a queue?	• Circular Array – Implement a queue as a circular array as follows: Use two index variables head and tail that contain the index of the next element to be removed and the next element to be added. After an element is removed or added, the index is incremented. ○ Blooms Taxonomy – Knowledge, Comprehension, Application, Analysis, Synthesis • Circular Array II – The queue of the previous program can fill up if more elements are added than the array can hold. Improve the implementation as follows. When the array fills up, allocate a larger array, copy the values to the larger array, and assign it to the elements instance variable. ○ Blooms Taxonomy – Knowledge, Comprehension, Application, Analysis, Synthesis

Suggested days of Instruction	**Curriculum Management System** **Grade Level/Subject:** **Advanced Computer Science (H)/ 11-12**	**Big Idea: Efficiency**	
		Topic: Introduction to Data Structures	
		Goal 10: The student will be able to understand and utilize a linked list and distinguish between concrete and abstract data types.	
	Objectives / Cluster Concepts / Cumulative Progress Indicators (CPI's) **The student will be able to:**	**Essential Questions** **Sample Conceptual Understandings**	**Instructional Tools / Materials / Technology / Resources / Learning Activities / Interdisciplinary Activities / Assessment Model**
			Assessment • Observation • Discussion • Quiz • Program Projects • Test

Suggested days of Instruction	**Curriculum Management System** **Grade Level/Subject:** **Advanced Computer Science (H)/ 11-12**	**Big Idea: Efficiency**	
		Topic: Advanced Data Structures	
		Goal 11: **The student will be able to understand and utilize a variety of advanced data structures.**	
	Objectives / Cluster Concepts / Cumulative Progress Indicators (CPI's) **The student will be able to:**	**Essential Questions** **Sample Conceptual Understandings**	**Instructional Tools / Materials / Technology / Resources / Learning Activities / Interdisciplinary Activities / Assessment Model**
12	11.1 understand set and map data type (4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2) 11.2 understand the implementation of hash tables (4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2) 11.3 program hash tables (4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2) 11.4 employ binary trees (4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2) 11.5 use tree sets and tree maps (4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2) 11.6 comprehend heap data structures (4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2)	• If Set s is a reference to some implementation of the Set interface, how could we add the element 1 to the set? • What do the values in a map form? • What is the purpose of a hash table? When would a programmer use one? • Refer to the hash function $f(x) = (3x + 4) \% 17$. What will the following integer values hash to? 0, 1, 17, 200 • How many cells will this hash table have? • Explain in detail what the binary tree can do. • Is it true that a node in a tree can be a child, parent, and sibling at the same time? Explain. • Refer to the following heap stored in the list form: 3 20 5 25 21 7 30 Draw the heap as a tree.	Instructional Tools/Materials/Technology/Resources • Cay Horstmann. *Java Concepts for AP Computer Science*. 5th ed. United States: John Wiley & Sons Inc., 2007. • Eclipse Programming Environment • Net Beans Programming Environment Learning Activities *Programs* *Students will use their understanding of advanced data structures to create the following programs:* • Binary Tree– Implement an application that builds a binary tree storing letters at each node. It then should print the letters in the tree using postorder traversal. Use TreeNode objects for each node. ○ Blooms Taxonomy – Knowledge, Comprehension, Application, Analysis, Synthesis • Family Tree – Design and implement an application that stores the programmer's own family tree in a binary tree. It should then print the tree. The root should be "the programmer" . ○ Blooms Taxonomy – Knowledge, Comprehension, Application, Analysis, Synthesis • Book – Write a class that represents a book. It should include instance variables that store

Suggested days of Instruction	**Curriculum Management System** **Grade Level/Subject:** **Advanced Computer Science (H)/ 11-12**	**Big Idea: Efficiency** **Topic: Advanced Data Structures** **Goal 11:** **The student will be able to understand and utilize a variety of advanced data structures.**	
	Objectives / Cluster Concepts / Cumulative Progress Indicators (CPI's) **The student will be able to:**	**Essential Questions** **Sample Conceptual Understandings**	**Instructional Tools / Materials / Technology / Resources / Learning Activities / Interdisciplinary Activities / Assessment Model**
	11.7 implement the priority queue data table (4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2) 11.8 understand how to use heaps for sorting (4.5.A.2, 4.5.C.2, 4.5.D.4, 4.5.F.2)	• How would a program implement the priority queue data table? • In a heap, what does the phrase bubbling up refer to?	properties of the book and methods that can be used to set the property values, and override the hashCode method in the class to create a hashcode form the book's properties. Write a driver program that creates some book objects, stores them, and prints them ○ Blooms Taxonomy – Knowledge, Comprehension, Application, Analysis, Synthesis *Essay* • Hash Tables – Research how computer security programmers use hash tables to secure the integretity of computer systems. Choose a specific hash table algorithm to report on. Write an essay on your findings. ○ Blooms Taxonomy – Knowledge, Comprehension, Application, Analysis, Synthesis, Evaluation Assessment • Observation • Discussion • Quiz • Program Projects • Test

Suggested days of Instruction	**Curriculum Management System** **Grade Level/Subject:** **Advanced Computer Science (H)/ 11-12**	**Big Idea:** **Nature**	
		Topic: Grid World	
		Goal 12: **The student will be able to apply prior knowledge to an in depth case study.**	
	Objectives / Cluster Concepts / Cumulative Progress Indicators (CPI's) **The student will be able to:**	**Essential Questions** **Sample Conceptual Understandings**	**Instructional Tools / Materials / Technology / Resources / Learning Activities / Interdisciplinary Activities / Assessment Model**
8	12.1 use and modify existing classes (8.2.B.1, 8.2.B.2)	• What methods does the bug have that the flower and rock do not have?	Instructional Tools/Materials/Technology/Resources • Cay Horstmann. *Java Concepts for AP Computer Science*. 5th ed. United States: John Wiley & Sons Inc., 2007. • Eclipse Programming Environment • Net Beans Programming Environment • AP Grid World Case Study • AP Grid World Case Study Narrative Learning Activities • Run the case study –Download the AP Grid World Case Study. Run the program. Observe and experiment with different objects and classes in the study. Understand the Bug class, Runner class, Grid Interface. Analyze the relationship between all of the classes and interfaces of the case study to understand why each are needed for the study to run. • Blooms Taxonomy – Knowledge, Comprehension, Application, Analysis • Extend the Bug class – Design and implement an extension of the Bug class. This extension can be any type of bug that you choose to create. It has to be a subclass of the Bug class. • Blooms Taxonomy – Knowledge, Comprehension, Application, Analysis, Synthesis • Exercises from the case study – Read the AP Grid World Case Study Narrative. Respond to
	12.2 understand intercommunicating objects (8.2.B.1, 8.2.B.2)	• Explain how all the classes in the case study interact with each other.	
	12.3 use inheritance (8.2.B.1, 8.2.B.2)	• How would you create a class jumper that extends bug? What new methods would you create, which would you override?	
	12.4 understand arrays (8.2.B.1, 8.2.B.2)	• All methods that return multiple objects return them in an ArrayList. Do you think it would be a better design to return them in an array? Explain your answer.	
	12.5 use of abstract classes (8.2.B.1, 8.2.B.2)	• During the design of the case study, some people thought it best to make the Actor class abstract. What do you think about this design? Give reasons why it should or should not be abstract.	
	12.6 use of data structures (8.2.B.1, 8.2.B.2)	• What is the time complexity (big-O) for the method `getValidAdjacentLocations` for a `BoundedGrid`?	

Suggested days of Instruction	**Curriculum Management System** **Grade Level/Subject:** **Advanced Computer Science (H)/ 11-12**	**Big Idea:** **Nature** **Topic: Grid World** **Goal 12:** **The student will be able to apply prior knowledge to an in depth case study.**	
	Objectives / Cluster Concepts / Cumulative Progress Indicators (CPI's) **The student will be able to:**	**Essential Questions** **Sample Conceptual Understandings**	**Instructional Tools / Materials / Technology / Resources / Learning Activities / Interdisciplinary Activities / Assessment Model**
			all questions that appear in the Narrative. • Blooms Taxonomy – Knowledge, Comprehension, Application, Analysis, Synthesis, Evaluation Assessment • Observation • Discussion • Quiz • Test • Program Projects

Suggested days of Instruction	**Curriculum Management System** **Grade Level/Subject:** **Advanced Computer Science (H)/ 11-12**	**Big Idea: Legal** **Topic: Computing in Context** **Goal 13: The student will be able to understand and explain legal issues surrounding technology.**	
	Objectives / Cluster Concepts / Cumulative Progress Indicators (CPI's) **The student will be able to:**	**Essential Questions** **Sample Conceptual Understandings**	**Instructional Tools / Materials / Technology / Resources / Learning Activities / Interdisciplinary Activities / Assessment Model**
4	13.1 comprehend system reliability (8.1.B.2) 13.2 be familiar with privacy issues (8.1.B.2) 13.3 understand legal issues and intellectual property (8.1.B.2) 13.4 be aware of social and ethical ramifications (8.1.B.2)	• Define in your own words system reliability. • State some privacy issues. Do you agree with these? • State and explain three legal issues involved with computers. • How have computers changed socialization? • What are the ethical issues involved with computer use?	Instructional Tools/Materials/Technology/Resources • The Net (1995) • Hackers (1995) • Internet Articles Learning Activities *Essays* • Invasion – After watching the Net, write an essay explaining how computers have allowed for a person's invasion of privacy. Then go on the Internet and research two real accounts of identity theft. Be sure to list these accounts in your essay along with the ramifications for the frauder and fraudee. ○ Blooms Taxonomy – Knowledge, Comprehension, Application • Legal – Research a computer crime that was committed recently on the Internet. Write an essay explaining what type of crime it was, ex. hacking, and the definition of that crime. Then explain what the person did to commit this crime. Last explain the ramifications the criminal received, and give a personal opinion about whether you feel the punishment was just given the crime. ○ Blooms Taxonomy – Knowledge, Comprehension, Application, Analysis, Synthesis, Evaluation

Suggested days of Instruction	**Curriculum Management System** **Grade Level/Subject:** **Advanced Computer Science (H)/ 11-12**	**Big Idea: Legal**	
		Topic: Computing in Context	
		Goal 13: The student will be able to understand and explain legal issues surrounding technology.	
	Objectives / Cluster Concepts / Cumulative Progress Indicators (CPI's) **The student will be able to:**	**Essential Questions** **Sample Conceptual Understandings**	**Instructional Tools / Materials / Technology / Resources / Learning Activities / Interdisciplinary Activities / Assessment Model**
			Assessment • Discussion • Essays

Suggested days of Instruction	**Curriculum Management System** **Grade Level/Subject:** **Advanced Computer Science (H)/ 11-12**	**Big Idea: Numbers**	
		Topic: Number Concepts	
		Goal 14: The student will be able to understand and utilize different number systems.	
	Objectives / Cluster Concepts / Cumulative Progress Indicators (CPI's) **The student will be able to:**	**Essential Questions** **Sample Conceptual Understandings**	**Instructional Tools / Materials / Technology / Resources / Learning Activities / Interdisciplinary Activities / Assessment Model**
3	14.1 convert decimal numbers to binary numbers (4.1.A.1) 14.2 understand and use the hexadecimal system (4.1.A.1)	• Why must all input and output be converted into binary code? • What is the hexadecimal system? Why is it used in programming?	Instructional Tools/Materials/Technology/Resources • Cay Horstmann. *Java Concepts for AP Computer Science*. 5th ed. United States: John Wiley & Sons Inc., 2007. • Internet • Eclipse Programming Environment • Net Beans Programming Environment Learning Activities *Programs* *Students will use their understanding of number systems to create the following programs:* • Binary – Create a program where the user can enter either a decimal number or a binary number. The program should recognize which type of number it is and converts decimal numbers to binary numbers and vice versa. ○ Blooms Taxonomy – Knowledge, Comprehension, Application, Analysis, Synthesis *Essay* • Hexadecimal – Research a specific area in computer programming where the hexadecimal system is useful. Explain in detail why the hexadecimal system is more advantageous in that scenario than the decimal system. Be sure to supply specific examples. ○ Blooms Taxonomy – Knowledge,

Suggested days of Instruction	**Curriculum Management System** **Grade Level/Subject:** **Advanced Computer Science (H)/ 11-12**	**Big Idea:** **Numbers**	
		Topic: Number Concepts	
		Goal 14: **The student will be able to understand and utilize different number systems.**	
	Objectives / Cluster Concepts / Cumulative Progress Indicators (CPI's) **The student will be able to:**	**Essential Questions** **Sample Conceptual Understandings**	**Instructional Tools / Materials / Technology / Resources / Learning Activities / Interdisciplinary Activities / Assessment Model**
			Comprehension, Application, Analysis Assessment • Observation • Discussion • Program Projects • Essay

Suggested days of Instruction	**Curriculum Management System** **Grade Level/Subject:** **Advanced Computer Science (H)/ 11-12**	**Big Idea: Business**	
		Topic: Independent Assignment	
		Goal 15: The student will be able to apply their previous knowledge to an independent assignment.	
	Objectives / Cluster Concepts / Cumulative Progress Indicators (CPI's) **The student will be able to:**	**Essential Questions** **Sample Conceptual Understandings**	**Instructional Tools / Materials / Technology / Resources / Learning Activities / Interdisciplinary Activities / Assessment Model**
8	15.1 evaluate a pre-existing software used in business (8.2.B.2) 15.2 create an original, improved program that completes the same task as the existing program that was researched (8.1.B.11, 8.2.B.3, 8.2.B.6)	• What software did you choose? In what type of industry is it used? • What are the improvements that you made on the existing program? How do you feel they will benefit the industry?	Instructional Tools/Materials/Technology/Resources • Internet • Eclipse Programming Environment • Net Beans Programming Environment Learning Activities *Essay* • Software – Choose an aspect of either business or education that uses a specific computer software program, ex. financial advising, banking, taxes, handling student grades, etc. Research the software in detail. Explain the pre-existing software you chose including what it does, its advantages, and its disadvantages. Discuss how you could make improvements to the existing program. o Blooms Taxonomy – Knowledge, Comprehension, Application, Analysis, Synthesis, Evaluation *Programs* • Original Program – You will create your own version of the program you researched. Your program should be able to accomplish the same tasks as the pre-existing software. Your program should also have the improvements you mentioned in the essay. o Blooms Taxonomy – Knowledge, Comprehension, Application, Analysis, Synthesis, Evaluation

Suggested days of Instruction	**Curriculum Management System** **<u>Grade Level/Subject</u>:** **Advanced Computer Science (H)/ 11-12**	**Big Idea: Business**	
		Topic: Independent Assignment	
		<u>Goal 15:</u> The student will be able to apply their previous knowledge to an independent assignment.	
	Objectives / Cluster Concepts / Cumulative Progress Indicators (CPI's) **The student will be able to:**	**Essential Questions** **Sample Conceptual Understandings**	**Instructional Tools / Materials / Technology / Resources / Learning Activities / Interdisciplinary Activities / Assessment Model**
			Assessment • Discussion • Program Projects • Essay

ADVANCED COMPUTER SCIENCE (H)

COURSE BENCHMARKS

1. The student will be able to understand how programming software works.
2. The student will be able to write classes and create objects.
3. The student will be able to effectively use primitive data, the String class, decision structures, and repetition structures.
4. The student will be able to use array and array lists to simplify coding and understand basic algorithms.
5. The student will be able to design efficient classes.
6. The student will be able to expand and create a hierarchy of classes
7. The student will be able to understand and use the recursive process and analyze pre-existing algorithms.
8. The student will be able to input and output to a file and handle exceptions.
9. The student will be able to comprehend object oriented programming and analyze the relationships formed between classes.
10. The student will be able to understand and utilize a linked list and distinguish between concrete and abstract data types.
11. The student will be able to understand and utilize a variety of advanced data structures.
12. The student will be able to apply prior knowledge to an in depth case study.
13. The student will be able to understand and explain legal issues surrounding technology.
14. The student will be able to understand and utilize different number systems.
15. The student will be able to apply their previous knowledge to an independent assignment.