



Syllabus

Dominican International School

AP Computer Science A

SY: 2019-20

Grade Level 11/12
1 Year

Teacher Tim Sampson
Email: tsampson@dishs.tp.edu.tw

Course Description

AP Computer Science A is equivalent to a first-semester, college level course in computer science. The course introduces students to computer science with fundamental topics that include problem solving, design strategies and methodologies, organization of data (data structures), approaches to processing data (algorithms), analysis of potential solutions, and the ethical and social implications of computing. For more details see the college board AP Computer Science A Course and Exam Description—Fall 2019

Requirements

The necessary prerequisites for entering the AP Computer Science A course include knowledge of basic algebra and experience in problem solving. A student in the AP Computer Science A course should be comfortable with functions and the concepts found in the uses of functional notation, such as $f(x) = 5x + 12$ and $f(x) = 5g(h(x))$. It is important that students and their advisers understand that any significant computer science course builds upon a foundation of mathematical reasoning that should be acquired before attempting such a course.

How this course can benefit students

The course emphasizes both object-oriented and imperative problem solving and design using Java language. These techniques represent proven approaches for developing solutions that can scale up from small, simple problems to large, complex problems. The AP Computer Science A course curriculum is compatible with many CS1 courses in colleges and universities.

Resources

- Online access to programming labs, homework,, e-text and supplemental videos:
<http://www.pearsonmylabandmastering.com/northamerica/myprogramminglab/>
- Our class site for discussion forums, collaboration and assignments.
<https://classroom.google.com>
- Supplementary documents, activities, labs and examples:
 - [The Java™ Tutorials](#)
 - [Gaddis Textbook Companion Website](#)
 - [Think Java, 2nd Edition](#) by Allen Downey and Chris Mayfield
 - [Practice-it University of Washington](#)
 - [Introduction to Computer Science using Java](#) by Bradley Kjell, Central Connecticut State University

- o <https://www.tutorialspoint.com/java>
- o <https://www.geeksforgeeks.org/java/>
- o <https://codingbat.com/java>
- o [AP CSA Labs](#) We will cover Chatbot, PixLab and Elevens in this order.
- Ethics in computing resources:
 - o [Bynum, Terrell, "Computer and Information Ethics](#)
<https://plato.stanford.edu/archives/win2016/entries/ethics-computer/>
 - o [Abelson, Ledeen, Lewis, "Blown to Bits: Your Life, Liberty, and Happiness After the Digital Explosion" <http://www.bitsbook.com/thebook/>](#)
 - o [Current media sources and Internet articles and blogs discussing ethical and social issues related to computer use.](#)

Textbooks

- ISBN: 978-1438005942 Teukolsky, R., & Barron's Educational Series, Inc. (2018). *AP Computer Science A*. [AP Computer Science A - Link](#)
- ISBN: 978-1260010336 5 Steps to a 5: AP Computer Science A 2018 2nd Edition by Dean R. Johnson , Carol A. Paymer , Aaron P. Chamberlain [5 Steps to a 5](#)
- ISBN: 978-0134447124 Starting Out with Java: Early Objects, 6th Edition, Tony Gaddis, Haywood Community College) *Plus Myprogramminglab with Pearson Etext – Access Card Package*. Pearson. [Starting Out with Java: Early Objects](#)
- Bynum, Terrell, "Computer and Information Ethics", *The Stanford Encyclopedia of Philosophy* (Winter 2015 Edition), Edward N. Zalta (ed.) <http://plato.stanford.edu/entries/ethics-computer>
- Abelson, H., Ledeen, K., & Lewis, H. R. (2008). *Blown to bits: Your life, liberty, and happiness after the digital explosion*. Upper Saddle River, NJ: Addison-Wesley. <http://www.bitsbook.com/excerpts/>

Videos

- There are companion videos for main topics in the text.
https://media.pearsoncmg.com/ph/esm/deitel/java_htpeo_10%20mpl/videonotes.html
- [Java Video Tutorial by Derek Banas](#)
- Supplemental videos may be added during the year depending on topic.
- IDE
- BlueJ Created by the University of Kent and supported by Oracle. This is a free program.

LTO's D'TORCH (Truthful, Organized, Reflective, Courageous and Helpful)

In CS classes the categories of the D'TORCH most practiced and assessed are:

- Organized - Students utilize Google Classroom to edit, submit and keep track of their assignments.
- Reflective - Students will regularly write activity reflections in their online journal.
- Helpful - Students are empowered to ask for and provide explanations and give examples to help classmates through particularly difficult problems.

Class Expectations

- Come to class on time and be prepared
- Have a positive attitude and be willing to learn.
- Respect yourself, others, and our school.
- Always complete your work and try your best.

- Actively participate, listen carefully, but don't speak out of turn.
- All assignments must be completed.

Homework and Quiz Rules

- All assignments must be turned in on the day they are due.
- 1 day late = Only 60%
- 2+ days late = Project-I & Only 60%
- If a student has been absent, it is his/her duty to find out what work is due, and hand it in a day later.
- All assignments must satisfactorily completed.
- If you are absent on the day of a quiz, you will only be able to get a maximum of 60%.

Classroom Rules

- All students are expected to follow the rules. Consequences will follow if rules are broken.
- Read and follow the standard school rules.
- Be on time and neatly dressed, in full school uniform.
- Speak in ENGLISH ONLY.
- Respect your teachers, fellow students and their property.
- Keep your seating space and classroom clean and neat.
- No eating or drinking in the ICT Labs.
- Ask permission to leave the class.
- Neither cheating nor copying in any form will be accepted. Anyone caught doing either during an activity, project or assessment will be given a zero.

Copying (plagiarism) is a serious offense and a form of theft. In certain cases, it is also a criminal offense. It is defined as taking words, phrasing, sentence structure, or any other element of the expression of another person's ideas, and using them as if they were your own. Plagiarism is a violation of another person's rights, whether the material stolen is great or small – it is not a matter of degree or intent. Plagiarism has serious consequences.

Any act of plagiarism will result in an automatic zero on the entire assignment.

Discipline

- Verbal warning
- Write-Up, entered into the discipline system and then referral to the Discipline Office.
- Parent-Teacher conference as required.

Course Design and Labs

This course is a four quarter course, with approximately 34 weeks of instructional time before the AP exams. All classes will regularly take place in the school's computer lab, except on occasions when students will be completing on off-line activities. Students will meet for six 45 minute class periods per week, which are arranged in two class blocks 3 times per week. A topic from the textbook is usually completed within two weeks. The time after AP exam will be used for enrichment activities and project work. In addition to the graded activities and labs there will be a test after each unit and also a final exam each quarter that covers all material covered up to that point in time.

Typically, class starts with a short introductory lecture and is followed by time allotted for students to complete practice questions and programming challenges, sometimes individually, pairs or groups. Some practice exercises and lab projects will be completed in [MyProgrammingLab](#), which is a browser based hands-on programming experience in an interactive online environment,. Some designing and implementation of Java classes will begin in offline activities individually, in pairs or groups. When there is a pair of group activity, students are allotted time to discuss the problem before implementing the program in the BlueJ IDE. These activities give students numerous opportunities to design and implement Java classes. [CR1] The total hands-on lab time for students will be a minimum of 20 hours of total instructional time, with labs distributed throughout the course. [CR6]

Note: This plan follows the APCS schedule as closely as possible, but has a few topics that are covered in a different order because of the materials we are using to provide a smooth path for the students. Notes are provided where there is a difference in the order of topics.

Q1

Unit 1: 2 weeks

During the first week of this AP unit, students will be providing time to get course materials and resources. In the second week, students will be introduced to the Java programming language and the use of classes, providing them with a firm foundation of concepts that will be leveraged and built upon in all future units. Students will focus on writing the main method and will start to call preexisting methods to produce output.

Topics

- ❑ 1.1 Why Programming? Why Java?
- ❑ 1.2 Variables and Data Types
- ❑ 1.3 Expressions and Assignment Statements
- ❑ 2.9 Using the Math Class
- ❑ 1.4 Compound Assignment Operators
- ❑ 1.5 Casting and Ranges of Variables

Resources

- ❑ textbook reading chapters 1 and 2
- ❑ Online reading in Think Java Chapters 1 and 2
- ❑ AP Review 5 Steps, Step 1, 4 along with Concept 0 Background on Software Development and Concept 1 Fundamentals of Programming

Activities and Assessments

- ❑ AP Personal Progress Check 1
- ❑ My Programming Lab Unit 1
- ❑ [Practice-It](#)
- ❑ CS Washington [expressions](#)
- ❑ Unit exam Textbook Ch 1 and 2.

Unit 2 Using Objects Part 1: 3 weeks

In the first unit, students used primitive types to represent real-world data and determined how to use them in arithmetic expressions to solve problems. This unit introduces a new type of data: reference data. Reference data allows real-world objects to be represented in varying degrees specific to a programmer's purpose. This unit builds on students' ability to write expressions by introducing them to Math class methods to write expressions for generating random numbers and other more complex operations. In addition, strings and the existing methods within the String class are an important topic within this unit. Knowing how to declare variables or call methods on objects is necessary throughout the course but will be very important in Units 5

and 9 when teaching students how to write their own classes and about inheritance relationships. Note: unit 5.8 Scope and Access is covered in our activities.

Topics

- ❑ 2.1 Objects: Instances of Classes
- ❑ 2.2 Creating and Storing Objects (Instantiation)
- ❑ 2.3 Calling a Void Method
- ❑ 2.4 Calling a Void Method with Parameters
- ❑ 2.5 Calling a Non-void Method
- ❑ 5.8 Scope and Access

Resources

- ❑ Textbook reading chapter 2 and 3
- ❑ Online reading in Think Java Chapters 3 and 4
- ❑ AP Review 5 Steps, along with Concept 2 Classes and Objects Basic Version

Activities and Assessments

- ❑ AP Personal Progress Check 2
- ❑ My Programming Lab Unit 2
- ❑ [Practice-It](#)
- ❑ Unit exam Textbook Ch 2.
- ❑ Unit exam Textbook Ch 3.
- ❑ Write a program that displays a pattern.
- ❑ Write a program that plays a word game with the user.
- ❑ Write a class named Employee
- ❑ Write a class named Car
- ❑ Write a Temperature class
- ❑ Design a Payroll class
- ❑ Design a WidgetFactory class
- ❑ Design a TestScore class

Unit 3 Boolean Expressions and if Statements: 2 weeks

Algorithms are composed of three building blocks: sequencing, selection, and iteration. This unit focuses on selection, which is represented in a program by using conditional statements. Conditional statements give the program the ability to decide and respond appropriately and are a critical aspect of any nontrivial computer program. In addition to learning the syntax and proper use of conditional statements, students will build on the introduction of Boolean variables by writing Boolean expressions with relational and logical operators. The third building block of all algorithms is iteration, which you will cover in Unit 4. Selection and iteration work together to solve problems.

Topics

- ❑ 3.1 Boolean Expressions
- ❑ 3.2 if Statements and Control Flow
- ❑ 3.3 if-else Statements
- ❑ 3.4 else if Statements
- ❑ 3.5 Compound Boolean Expressions
- ❑ 3.6 Equivalent Boolean Expressions
- ❑ 3.7 Comparing Objects

Resources

- ❑ Textbook reading Chapter 4 Decision Structures
- ❑ Online reading in Think Java Chapter 5: Conditionals and Logic
- ❑ AP Review 5 Steps Concept 3 The String Class

Activities and Assessments

- ❑ AP Personal Progress Check 3
- ❑ My Programming Lab Unit 4
- ❑ Practice-It! [If-else](#)
- ❑ Write a program that asks the user to enter a number
- ❑ Write a program class that asks the user to enter
- ❑ [AP Magpie Chatbot lab](#)

Q2

Unit 4: Iteration: 2 weeks

This unit focuses on iteration using while and for loops. As you saw in Unit 3, Boolean expressions are useful when a program needs to perform different operations under different conditions. Boolean expressions are also one of the main components in iteration. This unit introduces several standard algorithms that use iteration. Knowledge of standard algorithms makes solving similar problems easier, as algorithms can be modified or combined to suit new situations. Iteration is used when traversing data structures such as arrays, ArrayLists, and 2D arrays. In addition, it is a necessary component of several standard algorithms, including searching and sorting, which will be covered in later units.

Topics

- ❑ 4.1 while Loops
- ❑ 4.2 for Loops
- ❑ 4.3 Developing Algorithms Using Strings
- ❑ 4.4 Nested Iteration
- ❑ 4.5 Informal Code Analysis

Resources

- ❑ Textbook Chapter 5 Loops and Files
- ❑ Online reading in Think Java Chapter 6: Loops and Strings
- ❑ AP Review 5 Steps, Concept 4 The Math, Integer, and Double Classes

Activities and Assessments

- ❑ Practice-It! [for loops](#), [while loops](#)
- ❑ My Programming Lab Unit 5
- ❑ Unit exam Textbook Ch 5.
- ❑ Write a program that will predict the size of a population of organisms.
- ❑ Write a program class that asks the user for a positive integer.
- ❑ [AP Magpie Chatbot lab](#)

UNIT 5 Writing Classes: 2 weeks

This unit will pull together information from all previous units to create new, user-defined reference data types in the form of classes. The ability to accurately model real-world entities in a computer program is a large part of what makes computer science so powerful. This unit focuses on identifying appropriate behaviors and attributes of real-world entities and organizing these into classes. Students will build on what they learn in this unit to represent relationships between classes through hierarchies, which appear in Unit 9. The creation of computer programs can have extensive impacts on societies, economies, and cultures. The legal and ethical concerns that come with programs and the responsibilities of programmers are also addressed in this unit.

Topics

- ❑ 5.1 Anatomy of a Class
- ❑ 5.2 Constructors
- ❑ 5.3 Documentation with Comments
- ❑ 5.4 Accessor Methods
- ❑ 5.5 Mutator Methods
- ❑ 2.7 String Methods
- ❑ 5.6 Writing Methods
- ❑ 5.7 Static Variables and Methods
- ❑ 5.9 this Keyword
- ❑ 5.10 Ethical and Social Implications of Computing Systems

Resources

- ❑ Textbook Chapter 6: A Second Look at Classes and Objects
- ❑ Online reading in Think Java Chapter 9: Immutable Objects, Chapter 10: Mutable Objects, Chapter 11: Designing Classes
- ❑ AP Review 5 Steps, Concept 5 Data Structures
- ❑ AP Personal Progress Check 5

Activities and Assessments

- ❑ Practice-It! [Classes](#)
- ❑ My Programming Lab Unit 6
- ❑ Unit exam Textbook Ch 6.
- ❑ Write a class that has three overloaded static methods.
- ❑ Make a class with fields and methods.
- ❑ [AP Magpie Chatbot lab](#)

UNIT 6 Array: 2 weeks (including 2D and arrayList)

This unit focuses on data structures, which are used to represent collections of related data using a single variable rather than multiple variables. Using a data structure along with iterative statements with appropriate bounds will allow for similar treatment to be applied more easily to all values in the collection. Just as there are useful standard algorithms when dealing with primitive data, there are standard algorithms to use with data structures. In this unit, we apply standard algorithms to arrays; however, these same algorithms are used with ArrayLists and 2D arrays as well. Additional standard algorithms, such as standard searching and sorting algorithms, will be covered in the next unit.

Topics

- ❑ 6.1 Array Creation and Access
- ❑ 6.2 Traversing Arrays
- ❑ 6.3 Enhanced for Loop for Arrays
- ❑ 6.4 Developing Algorithms Using Arrays
- ❑ 7.5 Searching
- ❑ 7.6 Sorting

Resources

- ❑ Textbook Chapter 7 Arrays and the ArrayList Class
- ❑ Online reading in Think Java Chapter 7: Arrays and References and AP Review 5 Steps, Concept 6 Algorithms (Basic Version)
- ❑ AP Review 5 Steps, Concept 5 Data Structures

Activities and Assessments

- ❑ [AP Pix lab Student Guide](#)
- ❑ AP Personal Progress Check 6
- ❑ My Programming Lab Unit 2
- ❑ Practice-It! [Arrays](#)
- ❑ Write a class that has the following field: an array of doubles that stores values. It should also have methods that perform calculations. The program should get input from a user with validation.
- ❑ In a program, write a method that accepts two arguments: an array of integers and a number n. Your program should have a main method and store data in arrays.

UNIT 8 2D Array

In this section we swapped the order of units 7 and 8 to stay in sync with the way topics are presented in our textbook resources. In Unit 6, students learned how 1D arrays store large amounts of related data. These same concepts will be implemented with two-dimensional (2D) arrays in this unit. A 2D array is most suitable to represent a table. Each table element is accessed using the variable name and row and column indices. Unlike 1D arrays, 2D arrays require nested iterative statements to traverse and access all elements. The easiest way to accomplish this is in row-major order, but it is important to cover additional traversal patterns, such as back and forth or column-major.

Topics

- ❑ 8.1 2D Arrays
- ❑ 8.2 Traversing 2D Arrays

Resources

- ❑ Textbook Chapter 7 section 10 Two-Dimensional Arrays
- ❑ AP Review 5 Steps, Concept 5 Data Structures.

Activities and Assessments

- ❑ [AP Pix lab Student Guide](#)
- ❑ Practice-It!
- ❑ Write a program that lets the user enter values. The values should be stored in a two-dimensional array of doubles and outputted in a table format using /t.
- ❑ 2D Array Quiz

UNIT 7 ArrayList

As students learned in AP Unit 6, data structures are helpful when storing multiple related data values. Arrays have a static size, which causes limitations related to the number of elements stored, and it can be challenging to reorder elements stored in arrays. The ArrayList object has a dynamic size, and the class contains methods for insertion and deletion of elements, making reordering and shifting items easier. Deciding which data structure to select becomes increasingly important as the size of the data set grows, such as when using a large real-world data set. In this unit, students will also learn about privacy concerns related to storing large amounts of personal data and about what can happen if such information is compromised.

Topics

- ❑ 7.1 Introduction to ArrayList
- ❑ 7.2 ArrayList Methods
- ❑ 7.3 Traversing ArrayLists
- ❑ 7.4 Developing Algorithms Using ArrayLists

Resources

- ❑ Textbook Chapter 7 Section 13 The ArrayList Class
- ❑ Online reading in Think Java Chapter 13 Objects of arrays - ArrayList
- ❑ AP Review 5 Steps, Concept 5 Data Structures

Activities and Assessments

- ❑ Practice-It! [ArrayLists](#)
- ❑ [AP Pix lab Student Guide](#)
- ❑ Write a class with overloaded versions of all methods that work with int, float, double, and long arrays. Store data in arrays and output them on a separate line.
- ❑ Write a class with a constructor and use an ArrayList to store the values. Use a loop to display the values.
- ❑ Unit exam Textbook Ch 7 Arrays.
- ❑ AP Personal Progress Check 7

Christmas Break

Q3

Unit 2 Using Objects Part 2: 1 week

Topics

- ❑ 2.8 Wrapper Classes: Integer and Double
- ❑ 2.6 String Objects: Concatenation, Literals, and More
- ❑ 2.7 String Methods
- ❑ 4.3 Developing Algorithms Using Strings

Resources

- ❑ Textbook Chapter 8 Text Processing and More about Wrapper Classes
- ❑ Online reading in Think Java Chapter 14
- ❑ AP Review 5 Steps, Concept 7 Classes and Objects (Advanced Version)

Activities and Assessments

- ❑ [AP Elevens Lab](#)
- ❑ Unit exam Textbook Ch 8
- ❑ Write a class with a constructor that accepts a String object as its argument. The class should have a method that returns the number of vowels in the string, and another method that returns the number of consonants in the string.

UNIT 9 Inheritance

Creating objects, calling methods on the objects created, and being able to define a new data type by creating a class are essential understandings before moving into this unit. One of the strongest advantages of Java is the ability to categorize classes into hierarchies through inheritance. Certain existing classes can be extended to include new behaviors and attributes without altering existing code. These newly created classes are called subclasses. In this unit, students will learn how to recognize common attributes and behaviors that can be used in a superclass and will then create a hierarchy by writing subclasses to extend a superclass. Recognizing and utilizing existing hierarchies will help students create more readable and maintainable programs.

Topics

- ❑ 9.1 Creating Superclasses and Subclasses
- ❑ 9.2 Writing Constructors for Subclasses
- ❑ 9.3 Overriding Methods
- ❑ 9.4 super Keyword
- ❑ 9.5 Creating References Using Inheritance Hierarchies
- ❑ 9.6 Polymorphism
- ❑ 9.7 Object Superclass

Resources

- ❑ Textbook Chapter 9 Inheritance
- ❑ Online reading in Think Java Chapters 14 Extending classes
- ❑ AP Review 5 Steps, Concept 8 Inheritance and Polymorphism

Activities and Assessments

- ❑ Practice-It! [Inheritance](#)
- ❑ [AP Elevens Lab](#)
- ❑ AP Personal Progress Check 9
- ❑ Unit exam Textbook Ch 9.
- ❑ Design a class named Person with fields for holding data as Strings and a constructor that initializes all values, and mutator and accessor methods for every field.

UNIT 10 Recursion

Sometimes a problem can be solved by solving smaller or simpler versions of the same problem rather than attempting an iterative solution. This is called recursion, and it is a powerful math and computer science idea. In this unit, students will revisit how control is passed when methods are called, which is necessary knowledge when working with recursion. Tracing skills introduced in Unit 2 are helpful for determining the purpose or output of a recursive method. In this unit, students will learn how to write simple recursive methods and determine the purpose or output of a recursive method by tracing.

Topics

- ❑ 10.1 Recursion

Resources

- ❑ Resources include textbook Chapter 14 Recursion
- ❑ AP Review 5 Steps, Concept 10 Recursion

Activities and Assessments

- ❑ Practice-It! [Recursion](#)
- ❑ [AP Elevens Lab](#)
- ❑ Unit exam Textbook Ch 10
- ❑ Write a class with a constructor that accepts an array of test scores as its argument. Include a method that returns the average of the test scores. If any test score in the array is negative or greater than 100, the class should throw an `IllegalArgumentException`.

Q4

AP Mock Exam Preparation and Review: 3 Weeks

- ❑ Runestone [1.2. Pretest for the AP CS A Exam](#)
- ❑ AP Barron's Review

Mock Exams: 1 Week

During this time students are likely to be taking additional AP mock exams throughout the week. No assignments will be given to students during this time. Students can use this time for independent study.

AP Exam Preparation and Review: 1 Week

- ❑ 5 Steps to a 5 Review
- ❑ University of Washington Practice Exam
<https://courses.cs.washington.edu/courses/cse142/apcs/labs.shtml>

AP Exam

During this time students are likely to be taking additional AP exams throughout the week. No assignments will be given to students during this time. Students can use this time for independent study.

Appendix 1: APCS Chapter Mapping to Big Ideas

Big Ideas


For more details, please see the [AP Computer Science A Course and Exam Description—Fall 2019](#)

BIG IDEA 1: MODULARITY (MOD) Incorporating elements of abstraction, by breaking problems down into interacting pieces, each with their own purpose, makes writing complex programs easier. Abstracting simplifies concepts and processes by looking at the big picture rather than being overwhelmed by the details. Modularity in object-oriented programming allows us to use abstraction to break complex programs down into individual classes and methods.

BIG IDEA 2: VARIABLES (VAR) Information used as a basis for reasoning, discussion, or calculation is referred to as data. Programs rely on variables to store data, on data structures to organize multiple values when program complexity increases, and on algorithms to sort, access, and manipulate this data. Variables create data abstractions, as they can represent a set of possible values or a group of related values.

BIG IDEA 3: CONTROL (CON) Doing things in order, making decisions, and doing the same process multiple times are represented in code by using control structures and specifying the order in which instructions are executed. Programmers need to think algorithmically in order to define and interpret processes that are used in a program.

BIG IDEA 4: IMPACT OF COMPUTING (IOC) Computers and computing have revolutionized our lives. To use computing safely and responsibly, we need to be aware of privacy, security, and ethical issues. As programmers, we need to understand how our programs will be used and be responsible for the consequences.

Big Ideas	Unit 1	Unit 2	Unit 3	Unit 4	Unit 5	Unit 6	Unit 7	Unit 8	Unit 9	Unit 10
	Primitive Types	Using Objects	Boolean Expressions and if Statements	Iteration	Writing Classes	Array	ArrayList	2D Array	Inheritance	Recursion
Modularity MOD	✓	✓			✓				✓	
Variables VAR	✓	✓			✓	✓	✓	✓		
Control CON	✓	✓	✓	✓		✓	✓	✓		✓
Impact of Computing IOC					✓		✓			

AP Course at a Glance

UNIT 1 Primitive Types	UNIT 2 Using Objects	UNIT 3 Boolean Expressions and if Statements	UNIT 4 Iteration	UNIT 5 Writing Classes
~8–10 Class Periods 2.5–5% AP Exam Weighting	~13–15 Class Periods 5–7.5% AP Exam Weighting	~11–13 Class Periods 15–17.5% AP Exam Weighting	~14–16 Class Periods 17.5–22.5% AP Exam Weighting	~12–14 Class Periods 5–7.5% AP Exam Weighting
MOD 1.1 Why Programming? Why Java? VAR 1.2 Variables and Data Types CON 1.3 Expressions and Assignment Statements CON 1.4 Compound Assignment Operators CON 1.5 Casting and Ranges of Variables	MOD 2.1 Objects: Instances of Classes MOD 2.2 Creating and Storing Objects (Instantiation) MOD 2.3 Calling a Void Method MOD 2.4 Calling a Void Method with Parameters MOD 2.5 Calling a Non-void Method VAR 2.6 String Objects: Concatenation, Literals, and More VAR 2.7 String Methods VAR 2.8 Wrapper Classes: Integer and Double MOD 2.9 Using the Math Class	CON 3.1 Boolean Expressions CON 3.2 if Statements and Control Flow CON 3.3 if-else Statements CON 3.4 else if Statements CON 3.5 Compound Boolean Expressions CON 3.6 Equivalent Boolean Expressions CON 3.7 Comparing Objects	CON 4.1 while Loops CON 4.2 for Loops CON 4.3 Developing Algorithms Using Strings CON 4.4 Nested Iteration CON 4.5 Informal Code Analysis	MOD 5.1 Anatomy of a Class MOD 5.2 Constructors MOD 5.3 Documentation with Comments MOD 5.4 Accessor Methods MOD 5.5 Mutator Methods MOD 5.6 Writing Methods MOD 5.7 Static Variables and Methods VAR 5.8 Scope and Access VAR 5.9 this Keyword IOC 5.10 Ethical and Social Implications of Computing Systems

UNIT 6 Array		UNIT 7 ArrayList		UNIT 8 2D Array		UNIT 9 Inheritance		UNIT 10 Recursion																									
~6-8 Class Periods		10-15% AP Exam Weighting		~10-12 Class Periods		2.5-7.5% AP Exam Weighting		~10-12 Class Periods		7.5-10% AP Exam Weighting		~13-15 Class Periods		5-10% AP Exam Weighting		~3-5 Class Periods		5-7.5% AP Exam Weighting															
VAR 1 3	6.1 Array Creation and Access	VAR 1 3	7.1 Introduction to ArrayList	VAR 1 3	8.1 2D Arrays	MOD 1 3	9.1 Creating Superclasses and Subclasses	CON 1 5	10.1 Recursion	VAR 1 3	8.2 Traversing 2D Arrays	MOD 3 5	9.2 Writing Constructors for Subclasses	CON 1 5	10.2 Recursive Searching and Sorting	VAR 2 3 4	6.2 Traversing Arrays	VAR 2 3	7.2 ArrayList Methods	CON 3 4	7.4 Developing Algorithms Using ArrayLists	MOD 1 3	9.4 super Keyword	CON 2	7.6 Sorting	MOD 3 5	9.3 Overriding Methods	MOD 1 3	9.5 Creating References Using Inheritance Hierarchies	MOD 3 5	9.6 Polymorphism	MOD 1 3	9.7 Object Superclass
VAR 3 4	6.3 Enhanced for Loop for Arrays	VAR 2 3	7.3 Traversing ArrayLists	CON 3 4	7.5 Searching	CON 3 5	9.2	CON 2	7.7 Ethical Issues Around Data Collection	CON 3 5	9.3	9.4	CON 2	7.6	CON 3 5	9.5	9.6	CON 1 3	9.7	CON 1 3	9.1	9.2	CON 1 3	9.3	CON 1 3	9.4	CON 1 3	9.5	CON 1 3	9.6	CON 1 3	9.7	