Course Title: AP Computer Science (APCS)

Subject: Mathematics/Computer Science

Grade Level: 10-12

Duration: 36 weeks

Prerequisite: Computer Science 2 (CS2) with a grade of “A” or better

Elective or Required: Elective

Computer Science, Engineering, and Technology Mission Statement

Since computational thinking and problem solving are integral parts of our lives and 21st century learning, students must be actively involved in their Computer Science, Engineering, and Technology (CSET) education. The CSET curricula will emphasize thinking skills through a balance of computation, intuition, common sense, logic, design, analysis, and technology. Students will use a combination of technology and critical thinking to solve real-world problems. To achieve these goals, students will be taught a standards-based curriculum that is aligned with the New Jersey Curriculum Standards.

Course Description:

AP® Computer Science is an elective course that prepares students for the Advanced Placement Computer Science exam administered by the College Board. The course contains topics found in a college level Computer Science I course.

AP® Computer Science introduces students to object-oriented programming methodology with a concentration on problem solving and algorithm development using the Java programming language. A predefined subset of the Java standard library is utilized in this course. Some of the standard algorithms (for example, Binary and Sequential Search, Selection and Insertion Sort, Counting matches, Finding the highest and lowest score in a set) are explored. The College Board’s APCS-A labs: Magpie and Elevens are used as part of this course.
A variety of methods of instruction are used including but not limited to lecture, readings, videos, role playing, analysis of Java programs, and group and individual programming assignments. The course will meet for five 42-minute class periods per week for approximately 30 weeks prior to the AP® exam.

Lectures are interactive and almost always involve discussion and analysis of sample programs. For each unit of coverage, there are multiple programming assignments completed in class and assigned for homework. Students modify existing programs and also write classes and driver programs from scratch.

As a culminating activity following the AP® exam, students will work collaborative to design and develop a final project. The final project selected by the student will explore a technology, application development language, career or any other topic of interest to students and culminate in a well-documented original application or a research paper and a presentation to the class.

Author: Mayra Bachrach
Date Submitted: Summer 2016
Resources


Dr. Seuss, *The Cat in the Hat Comes Back*

Digital resources

Runestone Interactive *Java review for the APCS A exam website*, Barbara Ericson, Georgia Tech University:
http://interactivepython.org/runestone/static/JavaReview/index.html

Cisco Binary game website:
http://forums.cisco.com/CertCom/game/binary_game_page.htm

University of Washington’s Building Java Programs *Practice It* website:
https://practiceit.cs.washington.edu/login


*Codingbat* Java coding practice website: http://codingbat.com/

Teach APCS: http://teachapcs.com/

*Davidson College Computer Science Resources*, Ruth Hartsook, http://www2.davidson.k12.nc.us/hartsook/resources/index.htm

*Edhesive APCS YouTube Channel*:
https://www.youtube.com/playlist?list=PLGc4KcEkB4aCk1Mqcwx3fm2AqP4yJNYHD

Kjell, Bradley, Central Connecticut State University *Introduction to Computer Science using Java*, Java 1.5 version, January 2006,
http://chortle.ccsu.edu/cs151/cs151java.html

VisuAlgo Sorting Algorithm animation website: http://visualgo.net/sorting

College Board, *AP® Computer Science A labs, 2014*:
College Board, *Previous years APCS-A multiple choice and free response questions*,

Levine, David, *Role Play exercises*, St Bonaventure University, 2002,
http://www.cs.sbu.edu/dlevine/RolePlay/roleplay.html

Touretsky, David, *Common Lisp: A Gentle Introduction to Symbolic Computation (chapter 8: Recursion)*, Carnegie Mellon University, 1990,
http://www.cs.cmu.edu/~dst/

Association for Computing Machinery (ACM) code of Ethics, http://www.acm.org/


Glen Ridge High School Faculty pages: http://www.glenridge.org/grhs/faculty.html

**Software Tools**

- Google Chrome (or other web browser, such as Safari, Internet Explorer, etc)
- BlueJ
- JCreator
- Screencast-O-Matic (or other free screen recording software)
## Curriculum Standards

### Technology

<table>
<thead>
<tr>
<th>Standard</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.1.12.B.2</td>
<td>Apply previous content knowledge by creating and piloting a digital learning game or tutorial.</td>
</tr>
<tr>
<td>8.1.12.D.2</td>
<td>Evaluate consequences of unauthorized electronic access (e.g., hacking) and disclosure, and on dissemination of personal information.</td>
</tr>
<tr>
<td>8.1.12.D.4</td>
<td>Research and understand the positive and negative impact of one’s digital footprint.</td>
</tr>
<tr>
<td>8.1.12.D.5</td>
<td>Analyze the capabilities and limitations of current and emerging technology resources and assess their potential to address personal, social, lifelong learning, and career needs.</td>
</tr>
<tr>
<td>8.1.12.E.2</td>
<td>Research and evaluate the impact on society of the unethical use of digital tools and present your research to peers.</td>
</tr>
<tr>
<td>8.1.12.F.1</td>
<td>Evaluate the strengths and limitations of emerging technologies and their impact on educational, career, personal and or social needs.</td>
</tr>
<tr>
<td>8.2.12.B.3</td>
<td>Analyze ethical and unethical practices around intellectual property rights as influenced by human wants and/or needs.</td>
</tr>
<tr>
<td>8.2.12.D.6</td>
<td>Synthesize data, analyze trends and draw conclusions regarding the effect of a technology on the individual, society, or the environment and publish conclusions.</td>
</tr>
<tr>
<td>8.2.12.E.1</td>
<td>Demonstrate an understanding of the problem-solving capacity of computers in our world.</td>
</tr>
<tr>
<td>8.2.12.E.2</td>
<td>Analyze the relationships between internal and external computer components.</td>
</tr>
<tr>
<td>8.2.12.E.3</td>
<td>Use a programming language to solve problems or accomplish a task (e.g., robotic functions, website designs, applications, and games).</td>
</tr>
<tr>
<td>8.2.12.E.4</td>
<td>Use appropriate terms in conversation (e.g., troubleshooting, peripherals, diagnostic software, GUI, abstraction, variables, data types and conditional statements).</td>
</tr>
</tbody>
</table>

### 21st Century Life and Careers

<table>
<thead>
<tr>
<th>Standard</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.3.IT-PRG.4</td>
<td>Demonstrate the effective use of software development tools to develop software applications.</td>
</tr>
<tr>
<td>9.3.IT-PRG.5</td>
<td>Apply an appropriate software development process to design a software application.</td>
</tr>
<tr>
<td>9.3.IT-PRG.6</td>
<td>Program a computer application using the appropriate programming language.</td>
</tr>
<tr>
<td>9.3.IT-PRG.7</td>
<td>Demonstrate software testing procedures to ensure quality products.</td>
</tr>
<tr>
<td>9.3.IT-PRG.8</td>
<td>Perform quality assurance tasks as part of the software development cycle.</td>
</tr>
</tbody>
</table>

### 21st Century Life and Careers – Career Ready Practices

<table>
<thead>
<tr>
<th>CRP</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRP1</td>
<td>Act as a responsible and contributing citizen and employee.</td>
</tr>
<tr>
<td>CRP2</td>
<td>Apply appropriate academic and technical skills.</td>
</tr>
<tr>
<td>CRP4</td>
<td>Communicate clearly and effectively and with reason.</td>
</tr>
<tr>
<td>CRP6</td>
<td>Demonstrate creativity and innovation.</td>
</tr>
<tr>
<td>CRP8</td>
<td>Utilize critical thinking to make sense of problems and persevere in solving them.</td>
</tr>
<tr>
<td>CRP12</td>
<td>Work productively in teams while using cultural global competence.</td>
</tr>
</tbody>
</table>

### Mathematics

<table>
<thead>
<tr>
<th>Math Practices</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Math Practices.1</td>
<td>Make sense of problems and persevere in solving them.</td>
</tr>
<tr>
<td>Math Practices.2</td>
<td>Reason abstractly and quantitatively.</td>
</tr>
<tr>
<td>Math Practices.3</td>
<td>Construct viable arguments and critique the reasoning of others.</td>
</tr>
<tr>
<td>Math Practices.4</td>
<td>Model with mathematics.</td>
</tr>
<tr>
<td>Math Practices.5</td>
<td>Use appropriate tools strategically.</td>
</tr>
<tr>
<td>-----------------</td>
<td>--------------------------------------</td>
</tr>
<tr>
<td>Math Practices.6</td>
<td>Attend to precision.</td>
</tr>
<tr>
<td>Math Practices.7</td>
<td>Look for and make use of structure.</td>
</tr>
<tr>
<td>Math Practices.8</td>
<td>Look for and express regularity in repeated reasoning.</td>
</tr>
</tbody>
</table>

**English Language Arts**

<table>
<thead>
<tr>
<th>NJSLSA.R1</th>
<th>Read closely to determine what the text says explicitly and to make logical inferences and relevant connections from it; cite specific textual evidence when writing or speaking to support conclusions drawn from the text.</th>
</tr>
</thead>
<tbody>
<tr>
<td>NJSLSA.R2</td>
<td>Determine central ideas or themes of a text and analyze their development; summarize the key supporting details and ideas.</td>
</tr>
<tr>
<td>NJSLSA.R4</td>
<td>Interpret words and phrases as they are used in a text, including determining technical, connotative, and figurative meanings, and analyze how specific word choices shape meaning or tone.</td>
</tr>
<tr>
<td>RI.11-12.4</td>
<td>Determine the meaning of words and phrases as they are used in a text, including figurative, connotative, and technical meanings; analyze how an author uses and refines the meaning of a key term or terms over the course of a text (e.g., how Madison defines faction in Federalist No. 10).</td>
</tr>
<tr>
<td>RI.11-12.7</td>
<td>Integrate and evaluate multiple sources of information presented in different media or formats (e.g., visually, quantitatively) as well as in words in order to address a question or solve a problem.</td>
</tr>
<tr>
<td>RST.11-12.3</td>
<td>Follow precisely a complex multistep procedure when carrying out experiments, taking measurements, or performing technical tasks; analyze the specific results based on explanations in the text.</td>
</tr>
<tr>
<td>RST.11-12.4</td>
<td>Determine the meaning of symbols, key terms, and other domain-specific words and phrases as they are used in a specific scientific or technical context relevant to grades 11-12 texts and topics.</td>
</tr>
<tr>
<td>RST.11-12.10</td>
<td>By the end of grade 12, read and comprehend science/technical texts in the grades 11-CCR text complexity band independently and proficiently.</td>
</tr>
<tr>
<td>NJSLSA.W4</td>
<td>Produce clear and coherent writing in which the development, organization, and style are appropriate to task, purpose, and audience.</td>
</tr>
<tr>
<td>NJSLSA.W9</td>
<td>Draw evidence from literary or informational texts to support analysis, reflection, and research.</td>
</tr>
</tbody>
</table>
AP Computer Science

Unit 1: Intro to Computer Systems and Java

Approximate # Of Weeks: 1 week

Essential Questions:

- How are numbers represented in the hexadecimal, binary and octal number systems?
- How are numbers converted from one number system to another?
- What is the structure of a Java program?
- How is a Java class developed, compiled and executed?
- How are compiler error-messages interpreted and corrected in Java?

Upon completion of this unit students will be able to:

- Explain how computers represent numbers in binary and be able to convert and compare numbers in the binary, octal and hexadecimal number systems.
- Identify the basic structure of a Java class.
- Explain, identify and correct compiler errors.
- Explain the steps involved in program compilation and execution in Java.
- Install and configure Java and BlueJ on home computer.
- Write and execute a basic Java program.

Interdisciplinary Standards


Activities:

- Lecture and class discussion
- Activity: As part of the summer assignment, students will read notes on the Binary, Octal and Hexadecimal number systems and on how to convert numbers to the decimal, binary, octal and hexadecimal number systems. Students will write an explanation of the Number Systems Conversion process and demonstrate the conversion process by examples.
- Lab: Students will complete the interactive binary conversion problems (chapter 1: 1.1 - binary numbers, 1.2 – binary1 and 1.3 – binary2) on the Practice It website
- Class activity: Students will complete the Davidson College Computer Science Resources website’s lesson 4 on number systems conversion
- Class Activity: Working in pairs, students will review the conversion from decimal to binary and binary to decimal on the Cisco Binary game website
• Group project: Working in pairs, students will record a short video explaining how to convert numbers to different bases (binary, octal, hexadecimal).
• Online Review activity: *Java Basics* review on the *Java review for the APCS A exam* website.
• Readings and selected review exercises from Chapter 1 of Java Concepts: R1.1 thru R1.12 on pages 29-30.
• Lab: Java Concepts Chapter 1 Programming Exercises p1.1 thru p1.6 (Console output) on pages 30 and 31.
• Activity: Students will install and configure Java 1.5 (or higher) and BlueJ on their home computers.
• Online video for review and practice: Lessons 1a and 1b on the *Edhesive APCS YouTube channel* on Java basics and console output.

**Enrichment Activities:**

• Lab: Java Concepts chapter 1, programming exercises p1.7 and 1.8 on page 31 (Input and message dialogs).
• Project: Java Concepts chapter 1, programming project P1.1 on page 31 (Input and message dialogs).

**Methods of Assessments/Evaluation:**

• Unit quizzes.
• Unit test.
• Programming labs
• Projects
• Responses to discussion questions
• Verbal Assessment
• Think/Pair/Share
• Thumbs Up/Thumbs Down
• Exit slips

**Resources:**


**Online Resources:**

• Cisco Binary game site at [http://forums.cisco.com/CertCom/game/binary_game_page.htm](http://forums.cisco.com/CertCom/game/binary_game_page.htm)
• University of Washington Java Practice It website: [https://practiceit.cs.washington.edu/](https://practiceit.cs.washington.edu/)
• Runestone Interactive Java review for the APCS A exam site from Georgia Tech University: [http://interactivepython.org/runestone/static/JavaReview/index.html](http://interactivepython.org/runestone/static/JavaReview/index.html)
• Davidson College Computer Science Resources, Ruth Hartsook, [http://www2.davidson.k12.nc.us/hartsook/resources/index.htm](http://www2.davidson.k12.nc.us/hartsook/resources/index.htm)
- Edhesive APCS YouTube channel: https://www.youtube.com/playlist?list=PLGc4KcEkB4aCk1Mqcwx3fm2AqP4yJNYHD
- Teacher Webpage and YouTube channel
- Google Classroom/Drive class notes and exercises
AP Computer Science

Unit 2: Data types and Using Objects

Approximate # Of Weeks: 2.5 weeks

Essential Questions:

- What is a variable?
- What are the primitive data types?
- How are arithmetic operators represented and used in Java?
- What is an object?
- How are Java classes and objects related?
- How are objects constructed?
- What are methods, parameters and return values?
- How are the predefined classes in the Java API used?

Upon completion of this unit students will be able to:

- Explain, declare and use variables.
- Explain and use the primitive number types and the arithmetic operators.
- Explain and use object construction.
- Explain and use methods, parameters and return values.
- Explain, implement and use method overloading.
- Explain and use methods of the predefined classes in the Java API: System, Scanner, Rectangle, and String.
- Explain and use the Scanner and System classes for input and output.
- Explain the difference between a class, an object and an object reference.

Interdisciplinary Standards


Activities:

- Lecture and class discussion
- Readings and selected review exercises from Chapter 2 of Java Concepts: R2.1 thru R2.12 on pages 58-59.
- Class activity: (Object Role play) Students will role play objects to emphasize the difference between a class and an object, method invocation, parameter passing, return values and multiple objects of the same class (resource: David Levine’s First Day role play script from St. Bonaventure University).
- Lab: Java Concepts Chapter 2 Programming Exercises p2.1, p2.2 and p2.3 (Rectangle class), p2.9 and p2.10 (String manipulation) on pages 59, 60 and 61.
• Lab: BlueJ programming exercises on console output, using the Scanner class for input, passing parameters, declaring variables, arithmetic operations and String concatenation, including Piggy Bank and Temp Converter.

• Lab: Code Step by Step website exercises: expressionsNumbers1 and expressionsStrings1 under Expressions section (String and numbers concatenation).

• Lab: Students will complete Chapter 1 exercises (1.6 - legal identifiers, 1.7 – output syntax) from the Practice It website

• Lab: Exercise 4: Harmonic mean, Chapter 11 from Introduction to Computer Science Using Java website.

• Class Activity: Students will complete lesson 5, Math with ints and doubles, from the Davidson College Computer Science Resources website

• Online tutorials and videos for review and practice:
  o First 4 sections of the Variables lesson on Java review for the APCS A exam website
  o Lessons 3a and 3b on the Edhesive APCS YouTube channel and related exercises to review the basics of the Scanner class.
  o Lessons 4a, 4b and 4c on the Edhesive APCS YouTube channel and related exercises to review data types and the methods of the Scanner class.
  o Lessons 31a, 31b, 32a, 32b, 32c on the Edhesive APCS YouTube channel and related exercises to review method headers and passing parameters.
  o Lesson 8 on the Edhesive APCS YouTube channel and related exercises to review String concatenation Strings and numeric data types.
  o Lessons 35a, 35b and 35c on the Edhesive APCS YouTube channel and related exercises to review method overloading.

Enrichment Activities:

• Lab: Average Rainfall problem, Chapter 9B exercise 1 from Introduction to Computer Science Using Java
• Project: Java Concepts chapter 2, programming project P2.1 on page 61 (Gregorian calendar).

Methods of Assessments/Evaluation:

• Unit quizzes.
• Unit test.
• Programming labs
• Projects
• Responses to discussion questions
• Verbal Assessment
• Think/Pair/Share
• Thumbs Up/Thumbs Down
- Exit slips

**Resources:**


**Online Resources:**

- University of Washington Java Practice It website: [https://practiceit.cs.washington.edu/](https://practiceit.cs.washington.edu/)
- Runestone Interactive Java review for the APCS A exam site from Georgia Tech University: [http://interactivepython.org/runestone/static/JavaReview/index.html](http://interactivepython.org/runestone/static/JavaReview/index.html)
- Davidson College Computer Science Resources, Ruth Hartsook, [http://www2.davidson.k12.nc.us/hartsook/resources/index.htm](http://www2.davidson.k12.nc.us/hartsook/resources/index.htm)
- Edhesive APCS YouTube channel: [https://www.youtube.com/playlist?list=PLGc4KcEkB4aCk1Mqcwx3fm2AqP4yJNYHD](https://www.youtube.com/playlist?list=PLGc4KcEkB4aCk1Mqcwx3fm2AqP4yJNYHD)
- Teacher Webpage and YouTube channel
- Google Classroom/Drive class notes and exercises
AP Computer Science

Unit 3: Implementing Classes

Approximate # Of Weeks: 2 weeks

Essential Questions:

- How does a user-written class compare to a pre-defined Java class?
- What is Encapsulation?
- What is the public interface of a class?
- What is an instance variable and how is it used?
- How are local, instance and parameter variables similar and/or different?
- How do mutator and accessor methods compare?
- How are values passed to Java methods?
- What is the javadoc utility?

Upon completion of this unit students will be able to:

- Explain and create user-written classes.
- Explain and use Encapsulation.
- Explain, identify and declare instance fields of a class
- Explain and write constructors, accessor and mutator methods.
- Explain, identify and develop helper methods.
- Explain and determine return types and parameters for a method.
- Explain Java’s method of passing parameters to a method by value
- Explain and use a driver program to test a user-written class.
- Compare and contrast local variables, parameter variables and instance variables.
- Explain and use Javadoc comments.

Interdisciplinary Standards


Activities:

- Lecture and class discussion
- Readings and selected review exercises from Chapter 3 of Java Concepts: R3.1 thru R3.12 on pages 97-98
- Labs: Java Concepts Chapter 3 Programming Exercises p3.1, p3.2, p3.3 (BankAccount), p3.4 and p3.5 (Employee), p3.6 (Car), p3.7 (Student), p3.8 (Product), p3.9 (Circle), p3.10 (Square) on pages 98, 99.
- Labs: Practice It website Chapter 8 (Classes) exercises 8.5 through 8.11: Calculator Object, What is a field?, Name, accessorMutator, methodCallSyntax and getNormalReverseOrderName, exercises 8.17-8.19:
what is a Constructor? constructorPointErrors and constructorName, and exercises 8.23-8.25: publicVsPrivate, accessPrivateFields and setXYPoint.

- Class activity: Working in pairs, students will complete the Introduction to Object Method calls (lesson 10) on the Davidson College Computer Science Resources website.

- Class Activity: Working in pairs, students will complete lesson 17, Encapsulation – review the concept on the Davidson Computer Science Resources website.

- Project: Java Concepts chapter 3, programming project 3.2 (Greeter) on page 100.

- Online tutorial for review and practice: Object Oriented Concepts: Objects and Classes interactive lesson on the Java review for the APCS A exam website

Enrichment Activities:


- Project: Java Concepts chapter 3, programming project 3.1 (Enhanced BankAccount) on page 100.

Methods of Assessments/Evaluation:

- Unit quizzes.
- Unit test.
- Programming labs
- Projects
- Responses to discussion questions
- Verbal Assessment
- Think/Pair/Share
- Thumbs Up/Thumbs Down
- Exit slips

Resources:


Online Resources:

- University of Washington Java Practice It website: https://practiceit.cs.washington.edu/
- Runestone Interactive Java review for the APCS A exam site from Georgia Tech University: http://interactivepython.org/runestone/static/JavaReview/index.html
- Davidson College Computer Science Resources, Ruth Hartsook, http://www2.davidson.k12.nc.us/hartsook/resources/index.htm
- Teacher Webpage
- Google Classroom/Drive class notes and exercises
AP Computer Science

Unit 4: Fundamental Number Types and classes

Approximate # Of Weeks: 2.5 weeks

Essential Questions:

- What are the properties and limitations of the primitive number types in Java?
- How are the Math and String class used?
- How can overflow, loss of precision and round off errors be prevented?
- What is an escape sequence?
- What is a constant and why is it used?
- What is a static method?
- What algorithms are used to find the highest, lowest and average values in a set?

Upon completion of this unit students will be able to:

- Explain and apply the properties and limitations of the primitive number types.
- Explain and use arithmetic promotion, automatic type conversion and casting of primitives.
- Explain and use constants and avoid magic numbers.
- Explain and use the assignment, increment and decrement operators.
- Explain and use Integer division.
- Explain and apply the rules for operator precedence.
- Explain and call static methods.
- Explain, identify and avoid overflow, loss of precision and round off errors.
- Explain and use all the methods of the Math and String classes in the AP subset.
- Explain and apply the algorithm for finding the highest, lowest and average value in a set.
- Explain and use escape sequences.

Interdisciplinary Standards


Activities:

- Lecture and class discussion
• Readings and selected review exercises from Chapter 4 of Java Concepts: R4.1 thru R4.3, and R4.6 thru R4.12 and R4.16 on pages 141 -143.
• Labs: Java Concepts Chapter 4 Programming Exercises P4.1 (Cash Register), P4.3 (Power Generator), p4.4 (Pair of numbers), P4.5 (Dataset), P4.6 (Four integers, max/min), P4.13 (Quadratic Equation), P4.15 (Large letters), P4.16 (Print tree using Escape sequences) on pages 144-149.
• Labs: Practice It website
  o Chapter 1, Introduction to Java Programming, self-check exercises 1.9 through 1.15 on escape sequences.
  o Chapter 3, Parameters and Objects, self-check exercises 3.13, 3.15, 3.16 on the Math class
  o Chapter 3, Parameters and Objects, self-check exercises 3.19, 3.20, 3.21 to on the String class
• Lab: Code Step By Step website parameters and return section exercises: Compute Distance, getLastDigit, CircleArea, Average of 3
• Lab: Teach APCS website Pythagorean Theorem problem (Writing Functions section)
• Lab: Codingbat problems, helloName, makeAbba, makeTags, makeOutWords, firstHalf, withoutEnd, from the String-1 section
• Lab: Students will hand-write the solution to the APCS 2010 exam question 2, APLine and to the APCS 2015 exam question 2, Hidden Word
• Class Activity: Lesson 6, Using the Methods of the Math class, on the Davidson College Computer Science Resources website
• Class Activity: Lesson 16, String methods – learn and practice on the Davidson Computer Science Resources website
• Class Activity: Lessons 11 (Practice class methods) and 12 (All About Methods – learn and practice) on the Davidson Computer Science Resources website
• Online tutorials: Java Review for the APCS A exam website
  o Variables lesson (from Operators on)
  o Strings lesson
• Online videos/tutorials on Edhesive APCS YouTube channel for review and practice:
  o Lessons 2a and 2b to review escape sequences.
  o Lesson 6 to review Integer division and casting.
  o Lesson 7 to review the Modular Division (mod) operator.
  o Lessons 2a and 2b to review escape sequences.
  o Lesson 6 to review Integer division and casting.
  o Lessons 34a, 34b, 34c, 34d, 34e and 34f to review the return type of methods and the return statement.

Enrichment Activities:

• Project: Java Concepts chapter 4, programming exercise P4.18 (Easter Sunday) on page 148.
• Project: Java Concepts chapter 4, programming project P4.2 (Cash register – European market) on page 150.
Methods of Assessments/Evaluation:

- Unit quizzes.
- Unit test.
- Programming labs
- Projects
- Responses to discussion questions
- Verbal Assessment
- Think/Pair/Share
- Thumbs Up/Thumbs Down
- Exit slips

Resources:


Online Resources:

- University of Washington Java Practice It website: [https://practiceit.cs.washington.edu/](https://practiceit.cs.washington.edu/)
- Runestone Interactive Java review for the APCS A exam site from Georgia Tech University: [http://interactivepython.org/runestone/static/JavaReview/index.html](http://interactivepython.org/runestone/static/JavaReview/index.html)
- Davidson College Computer Science Resources, Ruth Hartsook, [http://www2.davidson.k12.nc.us/hartsook/resources/index.htm](http://www2.davidson.k12.nc.us/hartsook/resources/index.htm)
- Edhesive APCS YouTube channel: [https://www.youtube.com/playlist?list=PLGsKcEkB4aCk1Mqdx3fm2AqP4yJNYHD](https://www.youtube.com/playlist?list=PLGsKcEkB4aCk1Mqdx3fm2AqP4yJNYHD)
- Codingbat Java coding practice site: [http://codingbat.com/java](http://codingbat.com/java)
- Teacher Webpage and YouTube channel
- Google Classroom/Drive class notes and exercises
AP Computer Science

Unit 5: Decision statements

Approximate # Of Weeks: 2.5 weeks

Essential Questions:

- What is a decision statement?
- What are the different types of decision statements and when is each type applicable?
- How are relational and Boolean operators used in programming?
- What is short-circuiting?
- What are predicate methods?
- How are Strings ordered in the Unicode character set?
- How are Strings compared in Java?
- What is a truth table?
- How can Boolean expressions be simplified using De Morgan’s law?
- What does it mean when two reference variables are aliases?

Upon completion of this unit students will be able to:

- Explain and implement decision statements using if, if-else, if-else chains and nested if statements.
- Explain and use relational and boolean operators.
- Explain and apply short-circuiting.
- Explain, interpret and complete truth tables.
- Explain, interpret and apply De-Morgan’s law.
- Analyze and predict the outcome of program statements involving decision statements.
- Explain, identify, call and write predicate methods.
- Explain and be able to compare integers, floating point numbers, Strings and objects.
- Explain and use the .equals method.
- Explain the difference between the equals method and == when comparing object references.
- Explain and create aliases.
- Explain the lexicographic order of Strings and the Unicode character set.
- Explain and use the equals and compareTo methods to compare Strings.

Interdisciplinary Standards


Activities:

Last Updated: August 10, 2016
• Lecture and class discussion
• Labs: Java Concepts Chapter 6 Programming Exercises P6.1 (Quadratic equation), P6.2 (Card), P6.4 (Ordering 3 floating point numbers), P6.7 (Input Checker), P6.12 (Leap Year), P6.16 (Paycheck) on pages 225-228.
• Project: Design and develop the Rock-Paper-Scissor game.
• Class Activity: Lesson 7, Learn and Practice if, on the Davidson Computer Science Resources website
• Labs: Practice It website Chapter 4, Conditional Executions, problems: 4.2-logicExpressions1, 4.3 –ifStatementSyntax, 4.4-oops4-errors, 4.5-elseifMystery1, 4.6-elseifMystery2, 4.7-EvenOdd to review if else statements.
• Labs: Code Step By Step website exercises: percentageGrade (if else section) and isMultiple (parameters and return section).
• Labs: Introduction to Computer Science Using Java website Chapter 12 exercises: 4-Pie eating contest, 8-Matinee Movie tickets and 9-MidnightMadness and Chapter 14 exercises: 2-Tire Pressure and 3- More Tire Pressure
• Lab: Students will hand-write APCS 2010 exam question 2, APLine isOnline() method
• Lab: Students will hand-write modified version (no interface) of APCS 2008 exam question 4, Checker Objects.
• Online tutorial: Java Review for the APCS A exam website Conditionals lesson
• Review Online videos on Edhesive APCS YouTube channel
  o Lessons 11a, 11b, 12a, 12b to review if and if else statements.
  o Lessons 13a and 13b to review if else chains.
  o Lessons 14a, 14b and 14c to review the relational boolean operators (&& and ||) and truth tables.
  o Lessons 15 and 16 to review short circuiting (&& and ||) and DeMorgan’s law.
  o Lessons 21 and 22 to review Aliases and String comparisons.

Enrichment Activities:

• Project: Java Concepts chapter 6, programming project P6.1 (Combination lock) on page 229.
• Lab: Codingbat problems: cigarParty, dateFashion, squirrelPlay, caughtSpeeding and sortaSum, teenSum and maxMod5 under the Logic-1section, loneTeen under the Warmup-1section and withoutX and withoutX2 under the String-1 section.

Methods of Assessments/Evaluation:

• Unit quizzes.
• Unit test.
• Programming labs
• Projects
- Responses to discussion questions
- Verbal Assessment
- Think/Pair/Share
- Thumbs Up/Thumbs Down
- Exit slips

**Resources:**


**Online Resources:**

- University of Washington Java Practice It website: [https://practiceit.cs.washington.edu/](https://practiceit.cs.washington.edu/)
- Runestone Interactive Java review for the APCS A exam site from Georgia Tech University: [http://interactivepython.org/runestone/static/JavaReview/index.html](http://interactivepython.org/runestone/static/JavaReview/index.html)
- Davidson College Computer Science Resources, Ruth Hartsook, [http://www2.davidson.k12.nc.us/hartsook/resources/index.htm](http://www2.davidson.k12.nc.us/hartsook/resources/index.htm)
- Edhesive APCS YouTube channel: [https://www.youtube.com/playlist?list=PLGc4KcEkB4aCk1Mqcwx3fm2AqP4yJNYHD](https://www.youtube.com/playlist?list=PLGc4KcEkB4aCk1Mqcwx3fm2AqP4yJNYHD)
- Codingbat Java coding practice site: [http://codingbat.com/java](http://codingbat.com/java)
- Teacher Webpage and YouTube channel
- Google Classroom/Drive class notes and exercises
AP Computer Science

Unit 6: Iteration statements

Approximate # Of Weeks: 2.5 weeks

Essential Questions:

- What is a loop?
- What is a nested loop?
- How do while and for loops compare?
- What is the scope of a variable declared inside a loop?
- What are the common errors programmers make in using loops?

Upon completion of this unit students will be able to:

- Explain and implement iteration statements using while and for loops.
- Analyze and predict the outcome of program statements involving iteration statements.
- Explain and use the scope of variables declared inside loops.
- Explain, identify and avoid common errors with loops such as off-by-one and infinite loops.
- Compare while and for loops and determine when to use the structures.
- Explain, predict the outcome and implement nested loops.
- Explain, interpret and use boolean expressions to control loops.

Interdisciplinary Standards


Activities:

- Lecture and class discussion
- Readings and selected review exercises from Chapter 7 of Java Concepts: R7.1 thru R7.4, R7.8, and R7.13 on pages 269-270.
- Labs: Java Concepts Chapter 7 Programming Exercises P7.1 (Currency conversion), P7.5 (Fibonacci sequence), P7.11 (Prime numbers) on pages 270-273.
- Lab: Lewis, Loftus and Cocking programming exercises 3.4 (Sum of even), 3.5 (Print characters of String), 3.6 (number of odds, evens and zeroes) on page 183 and 3.11 (Hi-Lo game) and 3.13 (Draw triangle and diamond shapes using stars) on page 184-185
- Project: Write a Java class and driver that represent a grade calculator for a set of scores. Find the maximum, minimum and average grade in the set.
• Project: Java Concepts chapter 7, programming project 7.2 (Game of Nim) on page 276
• Project: Magpie activities 1 through 4 (APCS A labs)
• Labs: Code Step By Step website, loops section exercises: starsPrint, numberLoops1, computeSumOfDigits and rangeOfNumbers.
• Labs: Practice It website Chapter 4, Conditional Executions, problems: 4.17-cumulativeProduct, 4.19-precisionOfDoubles, and 4.23-lastFirst
• Labs: CodingBat problems: doubleChar, countHi and countDog in the String-2 section.
• Labs: Introduction to Computer Science Using Java website Chapter 16 exercises: 2 – sum of 1/n and Chapter 17 exercise: 3-Power of a number.
• Class Activity: Students will complete lesson 8, Code trace, and lesson 9, Practice loops, on the Davidson Computer Science Resources website
• Online tutorial: Java Review for the APCS A exam website Loops lesson
• Review Online videos On Edhesive APCS YouTube channel
  o Lesson 17a, 17b and 17c to review loops.
  o Lessons 18, 19 and 20 to review tracing through a loop and common types of loop problems on the APCS exam.

Enrichment Activities:

• Labs: CodingBat problem repeatSeparator in the String-2 section.
• Lab: Code Step By Step website loops section exercises: numberLoops2 and numberLoops3.
• Project: Java Concepts chapter 7, programming project 7.1 (Flesch Readability Index) on page 275

Methods of Assessments/Evaluation:

• Unit quizzes.
• Unit test.
• Programming labs
• Projects
• Responses to discussion questions
• Verbal Assessment
• Think/Pair/Share
• Thumbs Up/Thumbs Down
• Exit slips

Resources:

• Teukolsky, Roselyn, Barron’s Advanced Placement Computer Science A, 7th edition, Barron’s Educational Series, 2015

Online Resources:
• University of Washington Java Practice It website: 
  https://practiceit.cs.washington.edu/
• Step by Step website (Java section):  http://www.codestepbystep.com/
• Introduction to Computer Science Using Java chapter 16 at: 
  http://chortle.ccsu.edu/java5/Notes/chap16/progExercises16.html
• Introduction to Computer Science Using Java chapter 17 at: 
  http://chortle.ccsu.edu/java5/Notes/chap17/progExercises17.html
• Runestone Interactive Java review for the APCS A exam site from Georgia Tech University:  http://interactivepython.org/runestone/static/JavaReview/index.html
• Davidson College Computer Science Resources, Ruth Hartsook, 
  http://www2.davidson.k12.nc.us/hartsook/resources/index.htm
• Edhesive APCS YouTube channel: 
  https://www.youtube.com/playlist?list=PLGc4KcEkB4aCk1Mqcwx3fm2AqP4yJNYHD
• Codingbat Java coding practice site:  http://codingbat.com/java
• College Board, Previous years APCS-A multiple choice and free response questions, 
• College Board, AP® Computer Science A labs, 2014: 
• Teacher Webpage and YouTube channel
• Google Classroom/Drive class notes and exercises
AP Computer Science

Unit 7: Arrays and ArrayLists

Approximate # Of Weeks: 3.5 weeks

Essential Questions:

- How are lists represented in Java?
- How do Arrays and ArrayLists compare?
- How are the common list operations coded in Java (traverse, insert, replace and delete)?
- How are the common array algorithms for linear search, counting matches, calculating max and min and calculating the average programmed in Java?
- What are the Wrapper classes and why are they needed?
- What do the Integer.MAX_VALUE and Integer.MIN_VALUE constants represent?
- What are the ArrayIndexOutOfBoundsException and the IndexOutOfBoundsException exceptions?
- What is a two-dimensional array?
- How are operations on two-dimensional arrays programmed?

Upon completion of this unit students will be able to:

Students will be able to:

- Explain, construct and use an Array of primitives or reference types.
- Use a while or for loop to value and retrieve the values in an Array.
- Explain and use the enhanced for loop.
- Explain and use one dimensional Arrays
- Explain and use two dimensional Arrays
- Explain and use two dimensional Arrays as Arrays of arrays
- Explain and use partially filled Arrays.
- Explain, construct and use ArrayLists
- Explain and apply the properties and methods of ArrayLists.
- Explain and apply how to perform traversals, insertions, and deletions on Arrays and ArrayLists.
- Compare and contrast Arrays and ArrayLists.
- Explain the performance ramifications of Arrays and ArrayLists.
- Explain the difference between an Array of primitives and an Array of references.
- Explain and apply common Array/ArrayList Algorithms: Counting matches, linear search, finding the max, min and average.
- Explain and use the Wrapper classes (Integer and Double) and Auto-boxing.
- Explain, identify, avoid and correct the ArrayIndexOutOfBoundsException and IndexOutOfBoundsException exceptions.
Interdisciplinary Standards


Activities:

- Lecture and class discussion
- Readings and selected review exercises from Chapter 8 of *Java Concepts*: R8.3, R8.4, R8.7, R8.9, R8.11, R8.12, R8.16 on pages 312-315.
- Lab: Write a class that stores and manipulates a set of integers in an ArrayList using the Integer class and auto-boxing.
- Two-dimensional Array labs: Analyze and improve Tic-Tac-Toe class and driver from Java Concepts Chapter 8 pages 298 to 300, Java Concepts Chapter 8 Programming Exercises p8.18 and p8.19 on page 318, Magic Squares.
- Labs: *Introduction to Computer Science Using Java* website Chapter 47 exercise: 1 – sum of odd, even and all (1 dimensional array).
- Labs: *Introduction to Computer Science Using Java* website Chapter 49C exercise (modified to use non-ragged arrays): 1 – sum of all array elements, 2 – sum each row, 3 – sum of each column and 4 – max and min (two-dimensional array)
- Labs: *Teach APCS* website Factorial project and Array Fun project under Entering and Manipulating data (one dimensional array) and Seating Charts 2 project under Introducing classes (two dimensional Arrays).
- Labs: *Code Step By Step* java website arrays section exercises: `maxValue`, `findRange` and `getPercentEven`.
- Labs: *CodingBat* problems: `firstLast6`, `sameFirstLast`, `makePi`, `commonEnd`, `sum3`, `rotateLeft3`, `reverse3`, `sum2` in the Array-1 section.
- Labs: Students will hand-write an APCS previous exam question dealing with Arrays (see appendix 1 for question map)
- Labs: Students will hand-write an APCS previous exam question dealing with ArrayLists (see appendix 1 for question map)
- Labs: Students will hand-write an APCS previous exam question dealing with two-dimensional arrays (see appendix 1 for question map)
- Class Activity: Students will complete lesson 18 and 19, *Static Arrays*, on the Davidson Computer Science Resources website
- Project: Elevens Activity 1-4 – Card and Deck class (APCS A labs)
- Online tutorials: *Java Review for the APCS A exam* website
  - Arrays lesson
  - Lists lesson
  - Two-dimensional Arrays lesson
• Review Online videos On *Edhesive APCS YouTube channel*
  o Lessons 23a, 23b and 23c to review the concept of an array and how to access elements.
  o Lessons 24a, 24b and 24c to review traversing an array using a for loop.
  o Lessons 25a, 25b, 25c and 27 to review the common array algorithms.
  o Lessons 26a, 26b and 26c to review operations on arrays such as inserting an element and deleting an element.
  o Lessons 28a, 28b and 28c to review Array of objects.

**Enrichment Activities:**

• Lab: *Code Step By Step* website arrays section exercise: *matrixSum (two-dimensional arrays).*
• Project: Java Concepts chapter 8, programming project 8.1 (Poker Simulator) on page 319.

**Methods of Assessments/Evaluation:**

• Unit quizzes.
• Unit test.
• Programming labs
• Projects
• Responses to discussion questions
• Verbal Assessment
• Think/Pair/Share
• Thumbs Up/Thumbs Down
• Exit slips

**Resources:**


**Online Resources:**

• Code Step by Step website (Java section): [http://www.codestepbystep.com/](http://www.codestepbystep.com/)
• Introduction to Computer Science Using Java chapter 47 at: [http://chortle.ccsu.edu/java5/Notes/chap47/progExercises47.html](http://chortle.ccsu.edu/java5/Notes/chap47/progExercises47.html)
• Introduction to Computer Science Using Java chapter 49C at: [http://chortle.ccsu.edu/java5/Notes/chap49C/progExercises49C.html](http://chortle.ccsu.edu/java5/Notes/chap49C/progExercises49C.html)
• Runestone Interactive Java review for the APCS A exam site from Georgia Tech University: [http://interactivepython.org/runestone/static/JavaReview/index.html](http://interactivepython.org/runestone/static/JavaReview/index.html)
• Davidson College Computer Science Resources, Ruth Hartsook,
  http://www2.davidson.k12.nc.us/hartsook/resources/index.htm
• Edhesive APCS YouTube channel:
  https://www.youtube.com/playlist?list=PLGc4KcEkB4aCk1Mqcwx3fm2AqP4yJNYHD
• Codingbat Java coding practice site: http://codingbat.com/java
• Teach APCS http://teachapcs.com/
• College Board, Previous years APCS-A multiple choice and free response questions,
• College Board, AP® Computer Science A labs, 2014:
• Teacher Webpage and YouTube channel
• Google Classroom/Drive class notes and exercises
AP Computer Science

Unit 8: Designing Classes

Approximate # Of Weeks: 1.5 weeks

Essential Questions:

- How are pre-conditions, post-conditions and assertions addressed in program design?
- What are static or class variables?
- How and why are static methods written and used?
- What is the scope of a local, instance, class or parameter variable?
- What are cohesion and coupling?
- What are the side effects of modifying parameter variables?
- What are the guidelines for identifying classes during design?
- What are the guidelines for identifying and designing methods?
- How are packages used to organize classes?

Upon completion of this unit students will be able to:

- Explain and apply the guidelines for identifying classes during design.
- Explain and apply method decomposition.
- Explain, identify and use helper methods.
- Explain and apply cohesion and coupling during class design.
- Explain and identify immutable classes.
- Determine the mutator and accessor methods of a class.
- Explain, identify and manage the side effects of mutator methods.
- Explain and use Java's mechanism for passing parameters by value.
- Explain and use the difference between passing parameters of type primitive and reference.
- Explain, identify and comply with the pre-conditions and post-conditions of methods.
- Document pre-conditions and post-conditions using the javadoc utility.
- Explain, identify and use static methods and fields.
- Explain, analyze and use the difference in scope of instance fields, class variables, parameter variables and local variables.
- Explain, identify and avoid overlapping scopes.
- Understand and use packages to organize related classes.

Interdisciplinary Standards

Activities:

- Lecture and class discussion
- Labs: *Java Concepts* Chapter 9 Programming Exercises P9.1 (Coin) and P9.5 (static methods) on page 366.
- Lab: Modify the Employee class to include an employee number and use a static variable to assign the numbers in sequence.
- Lab: Students will trace the changes to parameter variables thru the execution of a program listing provided (Person).
- Lab: Students will complete the Static counter lab (static variables)
- Lab: Students will complete the Two-dimensional Arrays using static methods lab (static methods).
- APCS exam prep: Students will hand-write an APCS previous exam question (see appendix 1 for question map)
- Review Online videos On Edhesive APCS YouTube channel
  - Lessons 33a, 33b, 33c and 33d to review the difference between passing parameters of types primitive and Object.

Enrichment Activities:

- Project: Java Concepts chapter 9, programming project 9.1 (Paychecks) on page 369.

Methods of Assessments/Evaluation:

- Unit quizzes.
- Unit test.
- Programming labs
- Projects
- Responses to discussion questions
- Verbal Assessment
- Think/Pair/Share
- Thumbs Up/Thumbs Down
- Exit slips

Resources:


Online Resources:
• Edhesive APCS YouTube channel: https://www.youtube.com/playlist?list=PLGc4KcEkB4aCk1Mqcwx3fm2AqP4yJNYHD
• Teacher Webpage and YouTube channel
• Google Classroom/Drive class notes and exercises
AP Computer Science

Unit 9: Inheritance

Approximate # Of Weeks: 3 weeks

Essential Questions:

- What is inheritance and how is it implemented in Java?
- What is a super class or a subclass?
- What is a polymorphic reference?
- What are the rules for converting between super class and subclass types?
- What is dynamic binding?
- What methods are inherited by all classes from the Object class?
- How are abstract classes and methods used in a class hierarchy?
- What are the guidelines for determining the inheritance relationships between classes?
- How are UML diagrams used to document class hierarchies?
- What are the steps in the software development cycle?

Upon completion of this unit students will be able to:

- Explain and implement Inheritance by extending abstract and concrete classes.
- Explain the terms super class and subclass
- Explain and access inherited instance fields and methods.
- Explain and compare method overriding and method overloading.
- Explain and use the super keyword to call the constructor and overridden methods of a super class.
- Explain and use polymorphic references.
- Explain and use the rules for converting between super class and subclass types.
- Explain and use Dynamic Binding.
- Explain and use the Object class.
- Explain and override the toString and equals methods of Object.
- Explain, design and implement Abstract classes and methods.
- Explain and implement multiple inheritance using interfaces.
- Analyze the requirements of a group of classes to determine the inheritance relationships and where to use an abstract class, a concrete class or an interface in the hierarchy.
- Explain and apply the software life cycle.
- Explain, identify and implement relationships between classes (is-a, has-a).

Interdisciplinary Standards

- 8.2.12.E.1, 8.2.12.E.2, 8.2.12.E.3, 8.2.12.E.4, Math Practices 1-8, 21st Century Career Practices 1, 2, 4, 6, 8 and 12, 9.3.IT.PRG.4, 9.3.IT.PRG.5,
Activities:

- Lecture and class discussion
- Lab: *Java Concepts* Chapter 13 Programming Exercises P13.4 (Person hierarchy), P13.5 (Employee hierarchy), P13.6 (Worker hierarchy) on page 513.
- Lab: *Lewis, Loftus and Cocking* programming assignment 7.1 (Monetary Coin) on page 449.
- Lab: Analyze a driver program using polymorphic references to construct and manipulate objects from the BankAccount hierarchy and identify the classes being invoked for each method call.
- Lab: Modify the Student class to implement the .equals method of object based on the student GPA. Write a driver to test the class.
- Labs: *Practice It website* Chapter 9, Inheritance and Interfaces, problems: 9.3-SubClass syntax, 9.4-inheritanceVariableSyntax, 9.9-carTruck and 9.10-carTruck2
- Labs: *Code Step By Step* website Java Inheritance section exercises: inheritanceMystery1, inheritanceMystery2, inheritanceMystery3 and inheritanceMystery4.
- Project: Elevens Activities 6-9 – Board and AbstractBoard classes (APCS A labs).
- APCS exam prep: Students will hand-write an APCS previous exam question (see appendix 1 for question map)
- Online tutorial: *Java Review for the APCS A exam* website *Object Oriented Concepts* lesson (up to and including Abstract classes)

Enrichment Activities:

- Project: Java Concepts chapter 13, programming project 13.1 (Robots) on page 513

Methods of Assessments/Evaluation:

- Unit quizzes.
- Unit test.
- Programming labs
- Projects
- Responses to discussion questions
- Verbal Assessment
- Think/Pair/Share
- Thumbs Up/Thumbs Down
• Exit slips

**Resources:**


**Online Resources:**

• University of Washington Java Practice It website: [https://practiceit.cs.washington.edu/](https://practiceit.cs.washington.edu/)
• Code Step by Step website (Java section): [http://www.codestepbystep.com/](http://www.codestepbystep.com/)
• Runestone Interactive Java review for the APCS A exam site from Georgia Tech University: [http://interactivepython.org/runestone/static/JavaReview/index.html](http://interactivepython.org/runestone/static/JavaReview/index.html)
• Teacher Webpage and YouTube channel
• Google Classroom/Drive class notes and exercises
AP Computer Science

Unit 10: Testing, Debugging and Exceptions

Approximate # Of Weeks: 1 week

Essential Questions:

- How are unit testing, regression testing and integration testing used in the program development life cycle?
- How can a debugger and program traces be used to eliminate programming errors?
- How do syntax, logic and run-time errors compare?
- What are exceptions?
- How do checked and unchecked exceptions compare?
- What conditions are represented by the types of exceptions in the APCS API?

Upon completion of this unit students will be able to:

- Identify the three types of program errors: Compile-time, Run-time and Logic.
- Define and conduct unit testing, regression testing and integration testing
- Explain, identify and use typical, boundary and negative test cases.
- Explain debugging and use program traces to find program errors.
- Explain and use the various sources of test input.
- Explain and create suitable test cases.
- Explain and identify exceptions.
- Compare checked and unchecked exceptions.
- Identify, analyze and avoid common exceptions.

Interdisciplinary Standards


Activities:

- Readings and review exercises from Chapter 10 and 15 of Java Concepts: R10.1, R10.3, R10.6, R10.12 on page 403, R15.2 on page 571.
- Lecture and class discussion
- Lab: Java Concepts Chapter 10 Programming Exercises P10.12 (test cases for RootApproximator) on page 405
- Lab: Students will analyze 3 classes that generate exceptions to identify the exceptions, determine the causes and correct the errors.
• APCS exam prep: Students will hand-write an APCS previous exam question (see appendix 1 for question map) and answer multiple choice questions from APCS exam study guides.

Enrichment Activities:
• Project: Java Concepts chapter 10, programming project 10.1 (Unit Converter) on page 407

Methods of Assessments/Evaluation:
• Unit quizzes.
• Unit test.
• Programming labs
• Projects
• Responses to discussion questions
• Verbal Assessment
• Think/Pair/Share
• Thumbs Up/Thumbs Down
• Exit slips

Resources:
• Teukolsky, Roselyn, Barron’s Advanced Placement Computer Science A, 7th edition, Barron’s Educational Series, 2015

Online Resources:
• Google Classroom/Drive class notes and exercises
AP Computer Science

Unit 11: Interfaces and Polymorphism

Approximate # Of Weeks: 2.5 week

Essential Questions:

- What is polymorphism?
- How is a Java interface used to implement polymorphism?
- What are the rules governing conversion between class and interface types?
- How can polymorphic references of type interface be used to write reusable code?
- What are abstract methods?
- How do method overriding and method overloading compare?
- How is the Comparable interface used to compare objects?
- How is the ArrayList class related to the List interface?

Upon completion of this unit students will be able to:

- Explain the term polymorphism and its role in writing reusable classes.
- Explain and implement the Comparable interface and override the compareTo method.
- Explain the List interface and its relation to ArrayLists and other list data structures.
- Explain how interfaces are used to design generic, reusable software.
- Explain, design and implement user-written interfaces.
- Explain and use method overriding.
- Explain and apply the rules governing conversion between class and interface types.
- Explain and avoid a ClassCastException.

Interdisciplinary Standards


Activities:

- Lecture and class discussion
• Lab: Students will modify the Student class to implement the Comparable interface based on grade average and the Employee class to implement Comparable based on salary.
• Lab: Students will write the NumberGroup interface and the Range class which implements the interface.
• Lab: Students will write the Taxable interface and the Goods, Food, Toy and Book classes based on chapter 53 - Interfaces on the Introduction to Computer Science Using Java website.
• APCS exam prep: Students will hand-write an APCS previous exam question (see appendix 1 for question map)
• Online tutorial: Java Review for the APCS A exam website Interfaces and Polymorphism section of the Object Oriented Concepts lesson

Enrichment Activities:

• Project: Java Concepts chapter 11, programming project 11.2 (Board game manager) on page 441.

Methods of Assessments/Evaluation:

• Unit quizzes.
• Unit test.
• Programming labs
• Projects
• Responses to discussion questions
• Verbal Assessment
• Think/Pair/Share
• Thumbs Up/Thumbs Down
• Exit slips

Resources:

• Teukolsky, Roselyn, Barron’s Advanced Placement Computer Science A, 7th edition, Barron’s Educational Series, 2015

Online Resources:

• Introduction to Computer Science Using Java chapter 53 at: http://chortle.ccsu.edu/java5/Notes/chap53/ch53_1.html
• Runestone Interactive Java review for the APCS A exam site from Georgia Tech University: http://interactivepython.org/runestone/static/JavaReview/index.html
• Teacher Webpage and YouTube channel
• Google Classroom/Drive class notes and exercises
AP Computer Science

Unit 12: Recursion

Approximate # Of Weeks: 1.5 weeks

Essential Questions:

- What is recursion?
- What is infinite recursion?
- How does recursion compare to iteration?
- What are the performance implications of recursion?
- How can recursive code be analyzed to determine its outcome?

Upon completion of this unit students will be able to:

- Explain the concept of recursion.
- Explain and develop a recursive solution to a problem.
- Explain infinite recursion and identify and apply a base case to avoid it.
- Explain the relationship and difference between recursion and iteration.
- Explain and analyze the performance implications of recursive solutions.
- Analyze and trace thru the processing of a recursive method to predict its outcome.

Interdisciplinary Standards


Activities:

- Lecture and class discussion
- Class activity: Read The Cat in the Hat comes Back (Dr. Seuss) as an introduction to recursion.
- Class activity: Students will role play the recursive factorial algorithm from “Martin and the Dragon”, Common Lisp: A Gentle Introduction to Symbolic Computation.
- Labs: Java Concepts Chapter 18 Programming Exercises P18.1 (Reverse sentence, recursion), P18.3 (Reverse sentence, iteration), P18.6 (Largest element in Array, recursion) on page 695.
- Lab: Tracing recursion activity.
- Labs: Practice It website Chapter 12, Recursion, problems: 12.3-mystery1, 12.4-mystery2, 12.5-mystery3, 12.6-mysteryXY, 12.7-mystery4, 12.8-mystery5 and 12.9-mystery6.
• Class Activity: Students will complete lesson Recursion – learn and practice and Recursion – More practice on the Davidson Computer Science Resources website

• Class activity: Students will complete Recursion multiple choice questions from the Barron’s AP Computer Science A study guide and from previous APCS exams

• APCS exam prep: Students will hand-write an APCS previous exam question (see appendix 1 for question map)

• Online tutorial: Java Review for the APCS A exam website Recursion lesson

• Online videos: Edhesive APCS YouTube channel: Lessons 36a, 36b and 37

• Online video: University of Washington Recursive Tracing video tutorial

Enrichment Activities:

• Project: Students will work in pairs to complete Java Concepts chapter 18, programming project 18.13 (Towers of Hanoi) on pages 698 and 699

Methods of Assessments/Evaluation:

• Unit quizzes.

• Unit test.

• Programming labs

• Responses to discussion questions

• Verbal Assessment

• Think/Pair/Share

• Thumbs Up/Thumbs Down

• Exit slips

Resources:


• Teukolsky, Roselyn, Barron’s Advanced Placement Computer Science A, 7th edition, Barron’s Educational Series, 2015

• Dr. Seuss, The Cat in the Hat Comes Back

Online Resources:

• University of Washington Java Practice It website: https://practiceit.cs.washington.edu/

• Runestone Interactive Java review for the APCS A exam site from Georgia Tech University: http://interactivepython.org/runestone/static/JavaReview/index.html

• Davidson College Computer Science Resources, Ruth Hartsook, http://www2.davidson.k12.nc.us/hartsook/resources/index.htm


• Edhesive APCS YouTube channel: https://www.youtube.com/playlist?list=PLGc4KcEkb4aCk1Mqcx3fm2AqP4yJNYHD
- Teacher Webpage and YouTube channel
- Google Classroom/Drive class notes and exercises
AP Computer Science

Unit 13: Sorting and Searching

Approximate # Of Weeks:  2 weeks

Essential Questions:

- What is sorting?
- What is searching?
- What are the algorithms for Sequential search and Binary search?
- How does the performance of the Sequential and Binary search algorithms compare?
- What are the algorithms for Selection sort and Insertion sort?
- How does the performance of the Selection and Insertion sort algorithms compare?
- What is the algorithm for the MergeSort sort?
- How does the performance of MergeSort compare to the performance of Selection sort and Insertion sort?

Upon completion of this unit students will be able to:

- Explain and code the Sequential Search algorithm to search a list of primitives or objects.
- Explain and code the Binary Search algorithm to search a list of primitives or objects.
- Compare and contrast the Sequential search and Binary search algorithms.
- Explain and code the Selection sort algorithm to sort a list of primitives or objects.
- Explain and code the Insertion sort algorithm to sort a list of primitives or objects.
- Explain and code the Merge sort algorithm to sort a list of primitives or objects.
- Compare and contrast the Selection sort, Insertion sort and Merge sort algorithms.
- Explain, analyze and compare the performance of the search and sort algorithms.

Interdisciplinary Standards


Activities:

- Lecture and class discussion
• Lab: Students will write static Selection sort and Insertion sort methods to sort arrays of integers and objects and an ArrayList of objects.
• Lab: *Java Concepts* Chapter 19 Programming Exercises P19.1 and P19.2 (Selection sort), P19.4 (Merge sort), P19.12 (Sorting person) on page 738, 739.
• Class activity: Students will role-play sorting and searching an Array of numbers using the sort algorithms.
• Class activity: *Davidson Computer Science Resources* website lesson 24, Practice selection sort, lesson 26, practice Insertion sort and 28
• Class activity: Students will experiment with visual simulations of the sorting algorithms on the *VisuAlgo Sorting Algorithm animation* website.
• Lab: Students will complete multiple choice questions on sorting and searching from the *Barron's AP Computer Science A* study guide and from previous APCS exams.
• APCS exam prep: Students will hand-write an APCS previous exam question (see appendix 1 for question map)
• Online tutorial: *Java Review for the APCS A exam* website *Searching and Sorting* lesson

**Enrichment Activities:**

• Project: Working in pairs, students will design a manipulative to demonstrate one of the sorts or search algorithms. The students will present their project and how it can be used to simulate their selected algorithm.
• Lab: Analyze a ‘search and remove duplicates’ algorithm on an Array of integers to estimate the number of statements executed.

**Methods of Assessments/Evaluation:**

• Unit quizzes.
• Unit test.
• Programming labs
• Projects
• Responses to discussion questions
• Verbal Assessment
• Think/Pair/Share
• Thumbs Up/Thumbs Down
• Exit slips

**Resources:**

Online Resources:

- Runestone Interactive Java review for the APCS A exam site from Georgia Tech University: [http://interactivepython.org/runestone/static/JavaReview/index.html](http://interactivepython.org/runestone/static/JavaReview/index.html)
- Davidson College Computer Science Resources, Ruth Hartsook, [http://www2.davidson.k12.nc.us/hartsook/resources/index.htm](http://www2.davidson.k12.nc.us/hartsook/resources/index.htm)
- VisuAlgo Sorting Algorithm animation website: [http://visualgo.net/sorting](http://visualgo.net/sorting)
- Teacher Webpage and YouTube channel
- Google Classroom/Drive class notes and exercises
AP Computer Science

Unit 14: Review for APCS exam

Approximately two weeks will be allocated to reviewing for the AP exam. The review will include a mock AP exam which will count as a test grade in the 4th marking period.

Approximate # Of Weeks: 2 weeks

Upon completion of this unit students will be able to:

- Explain and complete selected free-response questions and multiple choice questions.
- Explain exam reader’s commentary and grading rubric on selected free-response questions.
- Explain AP free-response grading rubric.
- Achieve a grade of 3 or better on the APCS exam.

Interdisciplinary Standards


Activities:

- Reading and Exercises: College Board course description sample questions
- Reading and Exercises: College Board past exams free form questions
- Reading and exercises: College Board released multiple choice questions.
- Reading and Exercises: Barron’s Advanced Placement Computer Science Study Guide and Fran Tree’s Advanced Placement Computer Science Study Guide.
- Review of tests for all units.
- Online review video: University of Washington APCS Prep: Java String processing video

Methods of Assessments/Evaluation:

- Test (Mock APCS exam).
- Programming labs
- Responses to discussion questions
- Verbal Assessment
- Think/Pair/Share
- Thumbs Up/Thumbs Down

Last Updated: August 10, 2016
- Exit slips

**Resources:**


**Online Resources:**

- University of Washington APCS Prep: Java String processing video [https://www.youtube.com/watch?v=XCjcA56zufo&list=PL_bszZLe8OFnueQ6fn7wNqu87k3X2Nin&index=1](https://www.youtube.com/watch?v=XCjcA56zufo&list=PL_bszZLe8OFnueQ6fn7wNqu87k3X2Nin&index=1)
- Teacher Webpage and YouTube channel
- Google Classroom/Drive class notes and exercises
AP Computer Science

Unit 15: Ethical and Social Implications of Computer Use

Approximate # Of Weeks: 2 weeks

Essential Questions:

- What is the impact of computers and technology on the modern world and society?
- What are the social and ethical implications of Computing?

Upon completion of this unit students will be able to:

- Research and reflect on the social and ethical ramifications of advancements in computing.
- Analyze and discuss social and ethical ramifications of computer use and programming.

Interdisciplinary Standards


Activities:

- Class Activity: Visit the ACM web site. Read and discuss their published Code of Ethics.
- Project: Students will research a current event that involves computer use and write a short essay on the social and ethical implications to be considered. Class discussion on current events selected will be conducted.
- Students will select a topic in a chapter of Blown to Bits, present a summary (Powerpoint presentation) and lead a class discussion on the topic.
- Students will participate in a class discussion on social and ethical ramifications of computer use and programming.

Methods of Assessments/Evaluation:

- Verbal Assessment
- Think/Pair/Share
- Thumbs Up/Thumbs Down
- Exit slips
- Summary and reaction paper
- Class presentation
Online Resources:

- Association for Computing Machinery (ACM) code of Ethics, [http://www.acm.org/](http://www.acm.org/)
- Online magazines and newspapers
- Teacher Webpage and YouTube channel
- Google Classroom/Drive class notes and exercises
AP Computer Science

Unit 16: After the AP® Exam: Final project

After the AP® exam, students will work collaborative to design, develop and present an original application in a technology of their choice. As an alternative, students can research a current technology or historical event in computing, write a research paper, prepare a PowerPoint presentation and deliver a presentation to the class.

Approximate # Of Weeks: 4 weeks

Essential Questions:

- What are some of the computer languages or technologies used to develop modern day applications?
- What are some of the challenges encountered in developing computer applications?
- How can concepts and techniques learned in the course be applied to learning a new language and/or developing an application?
- What developments led to the creation of the computer?
- Who were the pioneers in the development of modern day computing?
- How is Computer Science applied in modern careers?

Upon completion of this unit students will be able to:

- Explain and use a computer language to design, develop and test an application.
- Explain the role that historical events played in the development of modern computing.
- Identify the pioneers who built on each other’s ideas to develop modern day computing.
- Identify some of the future uses of computers currently under research and development.
- Identify career options in the Computer Science field

Interdisciplinary Standards


Activities:

- Project: working in pairs or independently, students will select, design and complete one of the following final projects:
- Design, develop and test an application in Java. Students will present their completed project to the class.
- Learn a new programming language using internet tutorials and other resources. Design, develop and test an application in the selected language and present their completed project to the class.
- Research a current technology or historical event in computing, write a research paper, prepare a PowerPoint presentation and deliver the presentation to the class.
- Research career opportunities in the Computer Science field and write a paper describing the careers, professional and academic training required and other pertinent information. Students will prepare and deliver a PowerPoint presentation to the class.
- Students have the option of designing and completing a different project pre-approved by the teacher.

**Methods of Assessments/Evaluation:**

- Grading completed project as per teacher rubric
- Assessing daily progress and participation.

**Resources:**

- Internet tutorials, videos and programming resources
- Google classroom
- Search engine (Google)
- Teacher Webpage and YouTube channel
- Google Classroom/Drive class notes and exercises
## Appendix 1 – APCS Previous Exam question list by Unit/topic

<table>
<thead>
<tr>
<th>Unit/Topic</th>
<th>Related Previous APCS exam questions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Unit 7 - Arrays</strong></td>
<td>2004 question 4, Robot</td>
</tr>
<tr>
<td></td>
<td>2005 question 4 - Improve Grades</td>
</tr>
<tr>
<td></td>
<td>2006 question 3, Customer List</td>
</tr>
<tr>
<td></td>
<td>2007 question 1, Self-Divisor</td>
</tr>
<tr>
<td></td>
<td>2010 question 3, Trail</td>
</tr>
<tr>
<td></td>
<td>2011 question 1, Sound</td>
</tr>
<tr>
<td></td>
<td>2013 question 2, Token Pass</td>
</tr>
<tr>
<td><strong>Unit 7 - ArrayLists</strong></td>
<td>2004 question 1, Word List</td>
</tr>
<tr>
<td></td>
<td>2005 question 1, Hotel Reservation</td>
</tr>
<tr>
<td></td>
<td>2006 question 1, Daily Schedule</td>
</tr>
<tr>
<td></td>
<td>2007 question 3, Answer sheet</td>
</tr>
<tr>
<td></td>
<td>2008 question 1, Flight sheet</td>
</tr>
<tr>
<td></td>
<td>2008 question 2, String Coder</td>
</tr>
<tr>
<td></td>
<td>2009 question 1, Number Cube</td>
</tr>
<tr>
<td></td>
<td>2009 question 3, Battery Charger</td>
</tr>
<tr>
<td></td>
<td>2010 question 1, Cookie Order</td>
</tr>
<tr>
<td></td>
<td>2012 question 1, Climber</td>
</tr>
<tr>
<td></td>
<td>2013 question 1, SongList (List)</td>
</tr>
<tr>
<td></td>
<td>2014 question 1, Scrambler</td>
</tr>
<tr>
<td></td>
<td>2015 question 3, Sparse Array</td>
</tr>
<tr>
<td></td>
<td>2016 question 2, Log Messages</td>
</tr>
<tr>
<td></td>
<td>2016 question 4, String Formatter</td>
</tr>
<tr>
<td><strong>Unit 7 – Two Dimensional Arrays</strong></td>
<td>2011 question 4, Route Cipher</td>
</tr>
<tr>
<td></td>
<td>2012 question 4, Gray Image</td>
</tr>
<tr>
<td></td>
<td>2013 question 4, Sky View</td>
</tr>
<tr>
<td></td>
<td>2014 question 3, Seating chart</td>
</tr>
<tr>
<td></td>
<td>2015 question 1, Diverse Array</td>
</tr>
<tr>
<td></td>
<td>2016 question 3, Crossword</td>
</tr>
<tr>
<td><strong>Unit 9 - Inheritance</strong></td>
<td>2004 question 2, Pet</td>
</tr>
<tr>
<td></td>
<td>2005 question 2, Ticket Sales</td>
</tr>
<tr>
<td></td>
<td>2016 question 1, Random String Chooser</td>
</tr>
<tr>
<td><strong>Unit 11 - Interfaces</strong></td>
<td>2006 question 2, Taxable Items</td>
</tr>
<tr>
<td></td>
<td>2007 question 4, Game Design</td>
</tr>
<tr>
<td></td>
<td>2008 question 4, Checker Object</td>
</tr>
<tr>
<td></td>
<td>2009 question 4, Tile Game</td>
</tr>
<tr>
<td></td>
<td>2011 question 3, Fuel Depot</td>
</tr>
<tr>
<td></td>
<td>2012 question 3, Horse Barn</td>
</tr>
<tr>
<td></td>
<td>2014 question 4, Trio</td>
</tr>
<tr>
<td></td>
<td>2015 question 4, Number Group</td>
</tr>
</tbody>
</table>