# AP Computer Science A

## Course Information:

**Department:** Technology
**Course Title:** AP Computer Science A
**Length of Course:** Full year
**Grade Levels:** 9-12

## Instructor Information:

**Instructor:** Charlie Hyman
**Room:** 95
**E-mail:** hymanc@wlwv.k12.or.us
**Website:** http://www.etch.wlwv.k12.or.us/~whs_hyman/APCS/
**Resource Site:**
**Phone:** 503-673-7600
**Office Hours:** 7:45-8:30 or by appointment after school.

## Course Overview

AP Computer Science A is a college-prep level course which is intended to help students build a solid foundation in programming and working with complex data structures. The class introduces students to the concepts of object-oriented programming using the Java language. At the end of the year, students should have a strong understanding of programming basics, including data structures, control statements, objects, arrays, and sorting. Students should be able to analyze and understand existing code structures, adapt and modify code, and apply their analytical skills learned for Java to new situations and/or other programming and coding languages. Finally, students should be prepared and able to complete the AP Computer Science A Exam with a 3 or better.

Students will complete a variety of small skill-based exercises in the class, as well as two semester group projects and the collection of Grid World Case Studies from the College Board. Each student should have access to a computer every day and should learn and practice effective data-management techniques which will allow them to work on the same files at school and at home. Students will be expected to read from the textbook at home and complete programming exercises as homework. Lab time will be provided for in-class assignments and group projects but students are expected to program at home using open-source applications that can be downloaded from the class website.

### Essential Questions

- What is a computer program and what are its basic characteristics?
- What is object-oriented programming?
- What are the basic programming components of an object-oriented language?
- What is Java and how does it work?

- How do we apply analytical thinking skills to solve real-world problems through programming?
- What are our ethical and responsible obligations when working with technology?
- How do we work with other programmers to produce high-quality collaborative projects?
- How can we apply and extend skills learned with Java to other technologies?

### Ethics and Technology

In addition to the above, this class will emphasize the importance of ethical practices when working with technology. This is an important area, which deserves special attention, and will be woven throughout the framework of the course. My intention is to address each of these issues numerous times, touching on at least one issue per lesson, and to use case studies and examples as discussion points. The ethical areas that this class will look at are:

- Responsible and Ethical Use
- Privacy
- Economic and Legal Implications
- Safety and Harassment
- Intellectual Property Rights

### Grading

- Projects 15%
- Piazza entries & participation 10%
- Journal Entries: 10%
- Quizzes: 25%
- Tests/Exams: 40%

### Grading Information

- Grades will be updated regularly and should be accessible online at least every two weeks.
- Students seeking help can visit during office hours or by appointment.

### Course Expectations

- Students will be required to work on programming both in class and at home. Students should either have a google drive/dropbox account or usb flash drive for easy access to projects from anywhere.
- Be respectful-we will be sharing and reviewing each other's work and it is essential that we maintain a safe, positive, and constructive atmosphere at all times.
- Be on time-coming in late can be very disruptive. Please do not interrupt the class. If you come in late, it is your responsibility to make sure that you clear the tardy with the attendance office and bring me an admit slip.
- Be careful-abusing or misusing computers will not be tolerated.
- Do not download anything onto the computers unless you are explicitly told to do so.
- NO FOOD OR DRINK IN THE COMPUTER LAB-EVER!
- 20-Minute Rule-School policy requires that we remain in the classroom for the first 20 minutes of each period, so use the restroom during your passing time.
- Breaks-Only 2 students may leave the classroom at a time. No Exceptions! Students will use the sign out sheet and hall pass when leaving the classroom for any reason. Breaks must be limited to 5 minutes or less, anything more will be considered an absence.
- Watching videos, playing games, or other inappropriate use of computers during class time will result in a referral.

- Plagiarism of any kind, including attempting to pass off someone else's code as your own will result in an automatic zero for the assignment and a referral.
- Attendance is mandatory. Students who miss class for any reason are expected to make up missed work on their own time.

**Required Supplies**

- Students will provide their own book covers for textbooks.
- Students should have a three-ring binder for handouts, notes, and assignments.
- Students should have either a google drive/dropbox account or usb flash drive.
- Students will need to have reliable access to a computer outside of class.

## Course Resources

Horstmann, C. *Java Concepts Early Objects, 7th ed.* Hoboken, NJ: Wiley, 2014.

Sierra, K. and B. Bates. *Head First Java.* Sebastapol, CA: O'Riley Media, 2003.

*The Java Language Specification.* http://java.sun.com/docs/books/jls/

*Java 7 Platform Standard Edition 7.0 API Specification.*
http://docs.oracle.com/javase/7/docs/api/

http://docs.oracle.com/javase/tutorial/index.html

BlueJ Interactive Programming Environment. http://www.bluej.org/

College Board. *Computer Science A Course Description, Appendixes A and B.* York: College Entrance Examination Board, 2014.

## Methodology

The class will be broken up into units which are intended to address particular elements of the AP Computer Science A course. Each unit includes at least one programming project and a variety of smaller day-to-day exercises. The Hortsmann text, the major text for the class, will be included in each unit. Students will be expected to do assigned reading outside of class.

Students will work on individual small projects and will work together as members of small teams on some larger projects. Group work is intended to help students learn about proper documentation, creating good program flow-charts and JavaDocs, bug-testing, and interpreting and modifying code written by other programmers. The Semester projects will each be group projects and will include project proposals, planning, and implementation.

Each unit will end with a quiz that will address the concepts from that unit and provide practice for the AP Test. There will also be frequent short assessments and cumulative exams at the end of each quarter. The second-to-last unit, which focuses on review and preparation for the AP Test will include a practice test that will incorporate elements from each of the previous lessons.

Students will have individual access to computers every day and should spend a major portion of most class periods interacting with the BlueJ programming environment to implement the Java SE

JDK.  Students will also use three projects (Magpie, Picture, and Elevens), which will help them interact with existing code and work within a graphical environment.  By the end of the first semester, students should be comfortable enough with the code to adopt and adapt portions of it for their own purposes in the final programming project.

## Course Outline

### Fall Semester (18 weeks)

| Unit<br>(weeks) | Title, Topics, and Student Objectives | Resources, Assessments, and Strategies |
|---|---|---|
| 1<br>(3 weeks) | **Intro to programming and Java basics**<br>Object Oriented Programming<br>Design Structure<br>Properties<br>How Java Works<br>Java Code Structure<br>Pseudocode<br>Debugging<br>Compiling<br>Errors<br>Input/Output<br>Variables | Students will be introduced to programming structure and flowcharts.<br>Students will work on a variety of simple projects that are designed to illustrate programming methodology and OOP design.<br>Programming Project: Write **Hello World** program<br>Ethics Project: Acceptable Use Policy<br>**Hortsmann, ch 1**<br>This will also begin the Java programming portion of the class. Using BlueJ, students will become acquainted with the Java environment, writing pseudocode and comments, and creating simple code structures. Students will also learn how to debug common compiling errors.<br>Programming Project: Students will create a simple Login Project that takes user input and returns output to the user.<br>Java Interface Quiz |
| 2<br>(3 weeks) | **Objects Part 1—Implementing Classes**<br>Objects<br>Classes<br>Variables<br>Constructors<br>Reusing Code<br>Strings<br>Instance Fields<br>Return Types | **Hortsmann ch. 2 & 3**<br>Sierrra and Bates: Chapter 2, 3<br>Complete Chapter 2, 3 Exercises<br>Using BlueJ, students will begin by looking at existing objects and classes and discovering how to change and manipulate them.<br>Students will create a simple program that uses the pet class to create animal objects: cat, dog, fish, and bird.<br>Objects and Classes Quiz. |
| 3 | **Fundamental Data Types**<br>Primitives | **Hortsmann ch. 4** |

| | | |
|---|---|---|
| (3 weeks) | Number Types<br>Arithmetic Operators<br>Increment, Decrement<br>Strings<br>Concatenation<br>Escape<br>Constants | Students will explore data types and the difference between primitive data types and object.<br><br>Using BlueJ, students will create projects that will perform mathematical calculations and manipulate strings.<br><br>Students will use the Magpie lab to develop proficiency in the use and manipulation of strings<br><br>Q1 Test<br>Fundamental Data Types Quiz |
| 4<br>(4 weeks) | **Control Statements and Iteration**<br>if<br>if…then<br>if…else<br>if…then…else<br>for<br>while<br>switch<br>break<br>return statements | **Hortsmann, ch. 5 & 6**<br><br>Sierrra and Bates: Chapter 1<br>Complete Chapter 1 Exercises<br><br>Programming Project: Students will create a Zork Game<br><br>Ethics Case Study: User input and personal information.<br><br>Begin the Yahtzee Game<br><br>Loops Quiz. |
| 5<br>(4 weeks) | **Arrays**<br>List<br>One-dimensions Arrays<br>Two-dimensional Arrays<br>Array Lists<br>Inserting/Deleting | **Horstmann ch. 7**<br><br>Students will use the Picture lab to develop proficiency with two-dimensional arrays<br><br>Sierrra and Bates: Chapter 3, 9<br><br>Arrays vs. ArrayList Discussion (Sierrra and Bates: pp 137-139)<br><br>Programming Project: Dice Roller. Students will create a program that uses arrays and random number generation to roll dice and return a result.<br><br>Grid World Ch. 2<br><br>Continue Yahtzee Game<br><br>Arrays Quiz. |
| 6<br>(2 weeks) | **Behavior, Input/Output, and Exceptions**<br>Methods<br>Arguments and Return Types<br>Getters and Setters<br>Encapsulation<br>Input/Output | **Hortsmann, ch.11**<br><br>Sierrra and Bates: Chapter 4<br>Complete Chapter 4 Exercises<br><br>Review Computer Science A Appendix B |

| | | |
|---|---|---|
| | Exceptions | Complete Yahtzee Game<br>Behavior Quiz<br>Q2 Test |

**Spring Semester (18 weeks)**

| Unit (weeks) | Title, Topics, and Student Objectives | Resources, Assessments, and Strategies |
|---|---|---|
| 7<br>(4 weeks) | **Objects Part 2**<br>Object Creation<br>Object Interaction<br>Class Specification<br>Class Relationships<br>Class hierarchy<br>Class Design | **Hortsmann ch. 8**<br>Ethics Project: Ensuring a safe interactive environment.<br>Sierrra and Bates: Chapter 7, 8, 9<br>Complete Chapter 7, 8, 9 Exercises<br>Objects Quiz #2 |
| 8<br>(4 weeks) | **Interfaces and Inheritance**<br>Interface<br>Polymorphism<br>Inner Classes<br>Inheritance Hierarchies<br>Subclass Construction<br>Access Control | **Hortsmann ch. 9, 10**<br>Elevens lab<br>Sierrra and Bates: Chapter 12<br>Complete Chapter 12 Exercises<br>Interface Quiz |
| 9<br>(2 weeks) | **Recursion**<br>Recursion<br>Triangle Numbers<br>Permutations | **Hortsmann ch. 13**<br>Programming Project: Snake Game. Students will use a simple GUI and two-dimensional array to build a grid, create snake and food classes, and program a user interface.<br>Q3 Test<br>Recursion Quiz |
| 10<br>(2 weeks) | **Data Structures**<br>Sequential Search<br>Binary Search<br>Selection Sort<br>Insertion Sort<br>Merge Sort<br>HashSet<br>HashMap | **Hortsmann ch. 14, 15, 16**<br>Sierrra and Bates: Chapter 16<br>Complete Chapter 16 Exercises<br>Programming Project: Jukebox. Students will create a program that allows users to sort through a list of songs by title, genre, artist, or album. Users should also be able to add or delete songs. |

| | | |
|---|---|---|
| | | Sorting Quiz |
| 11<br>(3 weeks) | Review and preparation for AP Exam | Computer Science A Appendix A Review<br><br>Practice Exams<br>Free Response Questions |
| 12<br>(3 weeks) | Summative | Programming Project: Students build a simple tile-based game that relies on user-input, collision-detection, and simple physics.  The game will be object-oriented and will utilize two-dimensional arrays for world building.  A major goal will be to analyze how the Magpie, Elevens, and Picture lab  objects were set up and then apply similar concepts to creating their own game objects.<br><br>Final Exam<br><br>Ethics Case Study: Intellectual Property Rights |