

Mark Thompson
Phone: 229.225.5050
mathompson@tcjackets.net

AP Computer Science A
Course Syllabus
Room: PREP 115
2019 - 2020



Career Cluster: Information Technology
Pathway: Computer Science

AP CS A EXAM SCHEDULE: FRIDAY, MAY 8, 2020 12:00 PM (NOON)

Refer to the College Board course page for more information about Advanced Placement (AP) program exams and benefits. <https://apstudent.collegeboard.org/apcourse/ap-computer-science-a>. Please refer to the guidance or MERIT office for more information regarding testing fees and deadlines.

COURSE DESCRIPTION:

The course is based on numerous problem solving exercises, labs, and case studies, which require students to design and implement Java classes. ^[CR1] The course requires 40-50 hours of hands-on work in a computer lab. ^[CR6]

Objectives:

- Understand and apply the main principles of object-oriented software design and programming: classes and objects, constructors, methods, instance and static variables, inheritance, class hierarchies, and polymorphism
- Learn to code fluently in Java in a well-structured fashion and in good style; learn to pay attention to code clarity and documentation
- Learn to use Java library packages and classes within the scope of the AP Java subset
- Understand the concept of an algorithm; implement algorithms in Java using conditional and iterative control structures and recursion
- Learn to select appropriate algorithms and data structures to solve a given problem
- Compare efficiency of alternative solutions to a given problem
- Learn common searching and sorting algorithms: Sequential Search and Binary Search; Selection Sort, Insertion Sort, and Merge sort
- Understand one- and two-dimensional arrays, the List interface, and the ArrayList class, and use them appropriately in programming projects
- Acquire skills in designing object-oriented software solutions to problems from various application areas
- Discuss ethical and social issues related to the use of computers
- Prepare for the AP Computer Science A exam; meet all the curricular requirements defined by the College Board for this course.

COURSE CURRICULUM CONTENT:

Chapter numbers for readings and exercises refer to *Java Methods, 3rd AP Edition*. The labs, case studies, and projects proposed below come from *Java Methods* and serve only as examples of possible assignments; labs and assignments may vary.

Unit 1: An introduction to computers and software engineering (2 weeks)

1. Hardware, software and the Internet (Week 1; duration 1 week)

Elements of a computer system. How information is represented in computer memory. Binary and hex number systems and ASCII / Unicode. An introduction to the Internet.

Reading and exercises: Chapter 1.

Lab: Find and explore the home pages of some Internet and World Wide Web pioneers.

Lab: *Picture Lab*, Activities 1 and 2. [CR1, CR6]

2. An introduction to software engineering (Week 2; duration 1 week)

Getting familiar with the software development process. Compilers and interpreters. JDK tools (*javac*, *java*, *javadoc*). Running a Java program in a command-line environment (optional). Using an IDE. Java classes and source files. A brief introduction to OOP. Software engineer's Code of Ethics. [CR7]

Reading and exercises: Chapter 2 and Section 28.3

Lab: Compile and run simple programs (Hello World, Greetings) using command-line JDK tools or an IDE (Section 2.4).

Lab: Compile and run simple GUI applications (Section 2.6).

Unit 2: Syntax and objects (3 weeks)

3. Java syntax and style (Week 3; duration 1 week)

Syntax and style in a programming language. Comments. Reserved words and programmer-defined names. Statements, braces, blocks, indentation. Syntax errors, run-time errors, logic errors.

Reading and exercises: Chapter 3; Appendix A.

Lab: Correcting syntax errors and a logic error as an "adventure game" (Section 3.7). [CR6]

4. A first look at objects and classes (Weeks 4-5; duration 2 weeks)

Classes and objects. Classes and source files. CRC cards. Library classes and packages. The import statement. A first look at fields, constructors, and methods of a class. Inheritance.

Reading and exercises: Chapter 4 and *Elevens* Lab Student Guide, Activity 1

Lab: Design and implement Circle and Cylinder classes (Exercise 8, p. 92). [CR1, CR6]

Case study: *BalloonDraw* (Section 4.2). [CR1, CR6]

Case study and lab: *Balloons of All Kinds* (extend the *Balloon* class, coding constructors and overriding methods (Section 4.6). [CR1, CR6]

Lab: *Elevens*, Activity 1. [CR1, CR6]

Unit 3: Arithmetic, logic, and control statements (7 weeks)

5. Data types, variables, and arithmetic (Weeks 6-7; duration 2 weeks)

The concepts of a variable and a data type. Declarations of variables. Fields vs. local variables. The primitive data types: int, double and char. Literal and symbolic constants. Initialization of variables. Scope of variables. Arithmetic expressions. Data types in arithmetic expressions. The cast operator. The compound assignment (+=, etc.) and increment and decrement operators (++, --). Converting numbers and objects into strings. [CR5 (toString)] Math methods (abs, sqrt, pow, random). [CR5]

Reading and exercises: Chapter 5.

Lab: Exercises for Chapter 5 (for example, 16 - 19, p. 128). [CR1, CR6]

Lab: *Pie Chart* (Section 5.11). [CR1, CR6]

Lab: *Rainbow* (Exercise 27, p. 130). [CR1, CR6]

6. The if-else statement (Weeks 8-9; duration 2 weeks)

The if-else statement. Boolean expressions, the boolean data type, true and false values. Relational and logical operators. De Morgan's laws. Short-circuit evaluation. Nested if-else and if-else-if. *Case Study: Craps*. Elements of object-oriented design in *Craps*. The switch statement. enum data types.

Reading and exercises: Chapter 6.

Lab: Exercises for Chapter 6 (for example, 2-5, 10-12).

Lab: The Die [CR5 (random)] and CrapsGame classes for *Craps*: fill in the blanks and test in isolation (Section 6.9). [CR1, CR5, CR6]

Lab: Finishing and testing the *Craps* program (Section 6.12). [CR1, CR6]

Extra: codingbat.com *Logic-1* and *Logic-2*. [CR6]

7. Algorithms and iterations (Weeks 10-12; duration 3 weeks)

The concept of an algorithm. Properties of algorithms. Iterations. while, for, and do-while loops. break and return in loops. Nested loops. Euclid's GCF algorithm.

Reading and exercises: Chapter 7.

Lab: Exercises for Chapter 7 (for example, 1 - 11, 13, pp. 199-201). [CR6]

Lab: Print stars using iterations (Exercise 20, p. 204). [CR3, CR6]

Case study and lab: Euclid's GCF algorithm (Section 7.7 and Exercise 26 on p. 206).

Lab: *Perfect Numbers* (Section 7.8). [CR1, CR6]

Interlude: Ethical and social implications of computer use (Week 13, 1 week)

Student papers, presentations, and debates on ethical and social issues related to the use of computers and the Internet. [CR7]

Reading: Sections 28.3 - 28.5; current news and commentary in the online media.

Unit 4: Strings and arrays (4 weeks)

8. Strings (Week 14; duration 1 week)

String objects. Literal strings. Immutability. String methods. ^[CR5] Converting strings into numbers and numbers into strings. The Character class and its methods.

Reading and exercises: Chapter 8.

Lab: *Magpie*, Activities 1 and 2. ^[CR1, CR6]

Lab: *Lipograms* (Section 8.8). ^[CR1, CR6]

Extra: codingbat.com *String-1, String-2, String-3*. ^[CR6]

9. One-dimensional arrays (Weeks 15-16; duration 1.5 weeks)

One-dimensional arrays. Arrays as objects. Declaring and initializing. Indices. Length. `IndexOutOfBoundsException`. Traversals and the “for-each” loop. Inserting and removing elements.

Reading and exercises: Chapter 9.

Lab: *Fortune Teller* (Section 9.3). ^[CR1, CR6]

Lab: *Magpie*, Activity 5. ^[CR1, CR6]

Lab: Past free-response questions on arrays. ^[CR1, CR6]

Case study and lab: *The Sieve of Eratosthenes* (Section 9.8). ^[CR1, CR6]

Extra: codingbat.com: *Arrays-1, Arrays-2*. ^[CR2b]

10. Two-dimensional arrays (Weeks 16-17; duration 1.5 weeks)

Declaring and initializing two-dimensional arrays. Accessing the number of rows and columns. Traversals and nested “for-each” loops.

Reading and exercises: Chapter 9.

Lab: Past free-response questions on arrays. ^[CR1, CR6]

Lab: *Chomp* (Section 9.5).

Extra: codingbat.com: *Arrays-3*. ^[CR2b]

Unit 5: Classes and class hierarchies (6 weeks)

11. Details of defining classes and using objects (Weeks 18-19; duration 2 weeks)

Public and private fields and methods. Constructors and the `new` operator. References to objects. Calling methods and accessing fields. Passing parameters to constructors and methods. `return` statement. Overloaded methods. Static variables and methods.

Reading and exercises: Chapter 10.

Case study: the Fraction class (Sections 10.1 - 10.8). ^[CR1, CR6]

Case study and lab: *Snack Bar* (Section 10.9). ^[CR1, CR6]

Lab: *Snack Bar Continued* (Section 10.12). ^[CR1, CR6]

12. ArrayList (Weeks 20-21; duration 2 weeks)

ArrayList structure. The List interface. ArrayList's constructors and methods. Pitfalls. ArrayList vs. built-in arrays. [CR3]

Reading and exercises: Chapter 11.

Be Prepared, Section 2.5.

Lab: *Shuffler* (Section 11.4). [CR2b, CR4]

Lab: Creating an index for a document — using ArrayList and writing a subclass of ArrayList (Section 11.6). [CR2b, CR4]

Lab: Past AP free-response questions on ArrayList.

Lab: *ECG Analysis* (*Be Prepared*, Practice Exam 3, Question 1). [CR1, CR2b, CR6]

13. Class hierarchies, abstract classes, and interfaces (Weeks 22-23; duration 2 weeks)

Class hierarchies. Abstract classes. Invoking superclass's constructors and calling superclass's methods. Polymorphism. Interfaces.

Reading and exercises: Chapter 12.

Case study: *Happy Birthday Balloon* — writing a subclass of an abstract class Balloon (Section 12.5). [CR1, CR4, CR6]

Lab: *Baker's Dozen* (*Be Prepared*, Practice Exam 1, Question 2) — writing a class that implements an interface and a client class. [CR1, CR4, CR6]

Lab: *ChatBots* — implementing a small hierarchy of classes starting with an abstract class (*Be Prepared*, Practice Exam 4, Question 3). [CR1, CR4, CR6]

Lab: Past AP free-response questions on class hierarchies and polymorphism. [CR1, CR4]

Unit 6: Recursion, searching and sorting (4 weeks)

14. Recursion (Week 24; duration 1 week)

Recursive methods. Base case. Understanding and debugging recursive methods. When not to use recursion. [CR3]

Reading and exercises: Chapter 13 and Sections 23.3 - 23.5.

Lab: Chapter 13 exercises (for example, 6, 10 pp. 372 - 373). [CR1, CR6]

Lab: *The Tower Of Hanoi* (Section 23.5). [CR1, CR6]

15. Searching and sorting. Introduction to analysis of algorithms. (Weeks 25-27; duration 3 weeks)

Comparing objects. The equals method and the Comparable interface. Sequential and Binary Search. The number of comparisons required in Sequential and Binary Search. Selection Sort, Insertion Sort, and Mergesort. Comparison of efficiency of “quadratic” sorting algorithms (Selection Sort and Insertion Sort) vs. Mergesort.

Reading and exercises: Chapter 14.

Lab: Chapter 14 exercises.

Lab: *Keeping Things in Order* (Section 14.4). [CR1, CR2a, CR6]

Lab: *Benchmarks* (Section 14.9) — compares efficiency of several sorting algorithms. [CR2a]

Unit 7: Review (3 weeks)

16. Review and practice for the AP exam (Weeks 28-30; duration 3 weeks)

Quick reference (library classes and methods). ^[CR5] Past multiple-choice and free-response questions.

Reading: Be Prepared Chapters 1-5; *Be Prepared* Chapter 6 (past free-response questions and solutions), *Be Prepared* practice exams 1-5, *250 Multiple-Choice Computer Science Questions in Java*.

Unit 8: Enrichment (optional, duration varies)

17. Streams and files

Text and binary files. Streams vs. random-access files. Java I/O package. The Scanner class. Checked exceptions.

Reading and exercises: Chapter 15.

Lab: Choosing Words (Section 15.5).

Lab: Exercises and projects from exercises and the Test Package for Chapter 15.

18. Graphics and GUI

Computer graphics concepts. The Java Graphics class. GUI components and their events. Layouts. Handling mouse and keyboard events and images.

Reading and exercises: Chapters 16, 17, 18.

Lab: Pieces of the Puzzle (Section 16.7).

Programming project: Ramblecs (Section 17.6).

Lab: Slide Show (Section 18.7).

Unit 9: After the AP Exam (Duration varies)

Projects that demonstrate creative computer use.

Reading and exercises: Java Methods Chapter 28, “Computing in Context: Creative, Responsible, and Ethical Computer Use”, Section 28.2.

Other suggested activities: a team project to implement a game (for example, the Game of SET, <http://www.skylit.com/projects/> or the *Elevens* lab); or a potentially useful project for the school. ^[CR1, CR4, CR6]

TEXTS AND SUPPLEMENTARY MATERIALS:

- Litvin, Maria, and Gary Litvin. *Java Methods: Object-Oriented Programming and Data Structures, 3rd AP Edition*, Andover, Mass.: Skylight Publishing, 2015.
- Litvin, Maria, and Gary Litvin. *Be Prepared for the AP Computer Science Exam in Java, 6th Edition*, Andover, Mass.: Skylight Publishing, 2014.
- The College Board's *Maggie, Picture*, and *Elevens Labs* Student Guides.
- *CodingBat*: <http://codingbat.com/java>.
- Litvin, Maria, and Gary Litvin. *250 Multiple-Choice Computer Science Questions in Java*. Andover, Mass.: Skylight Publishing, 2008.
- Current media sources and Internet articles and blogs discussing ethical and social issues related to computer use.
- Google Classroom an Online Learning Management System (LMS) will be used for managing assignments.
- Students should bring a writing instrument and notebook to class each day

GRADING CATEGORIES:

Daily Grades/In Class Assignments	10%
Tests and Quizzes	45%
Projects/Lab Work	25%
Benchmark (Final)	20%

GRADING POLICY:

The grading scale is as follows: A = 90 – 100, B = 80 – 89, C = 70 – 79, below 70 is failing.

CLASSWORK:

The course requires 40-50 hours of hands-on work in a computer lab. Classroom assignments are designed to be completed during class time. Classwork must be completed and submitted during class. Late work is not accepted. Making up work for excused absences is the responsibility of the student. Students should consult Google Classroom and inform the teacher to make up assignments for excused absences.

HOMEWORK:

Homework assignments reinforce concepts introduced in class. Students will read, study, and complete exercises for homework. Homework must be submitted within the first five minutes of class on the day it is due. Late work is not accepted.

CLASSROOM RULES/CONDUCT:

As part of the P.R.E.P. Academy, the Business and Computer Science Department focuses on professionalism, accountability, responsibility, self-discipline and similar work ethics that are expected behaviors in a business environment. Therefore, each student is expected to conduct himself/herself in a professional manner by avoiding the following infractions: (1) unnecessarily stopping the teacher from

teaching, (2) hindering other students from learning, and (3) engaging in behavior that is not in the best interest of the class. To insure that a positive learning atmosphere is maintained, the teacher will enforce the discipline procedures outlined in the *Thomas County Central High School Parent-Student Handbook*.

CONSEQUENCES FOR MISCONDUCT:

- 1st Offense: Verbal Warning. Documented.
- 2nd Offense: Call Parent or Guardian. Documented.
- 3rd Offense: Teacher Detention before or after school. Documented.
- 4th Offense: Disciplinary write-up to the grade-level administrator.

COMPUTER USE:

Students will be required to access the Internet daily assignments and projects. Each student must have an Acceptable Use Policy (AUP) on file at the school. All policies in the AUP will be followed.

Students should use the internet when instructed for classroom purposes only. Students who violate the AUP will receive a discipline referral and may have their computer privileges revoked.

FUTURE BUSINESS LEADERS OF AMERICA (FBLA):

FBLA is a co-curricular student organization that plays an integral part in the components of the Business & Technology course standards. FBLA activities are incorporated throughout this course and the rest of the Business and Computer Science courses. Students are strongly urged to join FBLA (\$25) to benefit from the wealth of opportunities the organization has to offer.

END OF PATHWAY ASSESSMENT

Students are encouraged to select a pathway beginning in ninth grade that is connected to their college and career goals. This course is one of three courses required to complete the Computer Science or Programming pathway in the CTAE department. At the conclusion of the third pathway course, students will be required to take an End of Pathway Assessment. This assessment provides students an opportunity to demonstrate what they have learned by completing an on-line, nationally recognized exam (Microsoft Software Development Fundamentals Certification 98-361). Students who complete a pathway and earn an industry credential by passing the assessment will receive a graduation cord to signify their achievement.

CAREER OPPORTUNITIES

Computer Science Principles, as part of the Computer Science and Programming pathways, can help prepare students for careers in Information Technology, Management Information Systems, Computer Programming, and Computer Engineering.

Please read the following statements, print your name, sign, and fill out the information below.

As the **student**, I have read the AP Computer Science A syllabus and understand the expectations and requirements of the course. I also agree to follow the rules in Mr. Thompson's classroom.

Student Printed Name	Student Signature	Date
-----------------------------	--------------------------	-------------

As the **Parent/Guardian**, I have read the AP Computer Science A syllabus and understand the expectations and requirements of the course. I expect my TCCHS student to follow the rules in Mr. Thompson's classroom.

Parent/Guardian Printed Name	Parent/Guardian Signature	Date
-------------------------------------	----------------------------------	-------------

Parent Contact Information: *Please indicate the preferred phone number.*

Home: _____ Best time to call: _____

Work: _____ Best time to call: _____

Cell: _____ Best time to call: _____

Email: _____