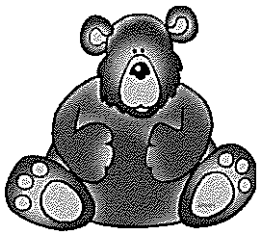
Mrs. Shirl Williams
Email: shirl.williams@hcbe.net
478-988-6340 ext 32830



AP Computer Science A
Syllabus 2018 - 2019

The Course

This is an introductory course in computer science. Because the development of computer programs to solve problems is a skill fundamental to the study of computer science, a large part of the course is built around the development of computer programs or parts of programs that correctly solve a given problem. The course also emphasizes the design issues that make programs understandable, adaptable, and, when appropriate, reusable. At the same time, the development of useful computer programs and program modules is used as a context for introducing other important concepts in computer science, including the development and analysis of algorithms, the development and use of fundamental data structures, and the study of standard algorithms and typical applications. In addition, an understanding of the basic hardware and software components of computer systems and the responsible use of these systems are integral parts of the course.

Goals

The goals of AP Computer Science A are comparable to those in the introductory sequence of courses for computer science majors offered in college and university computer science departments. It is not expected, however, that all students in an AP Computer Science course will major in Computer Science at the university level. An AP Computer Science course is intended to serve both as an introductory course for computer science majors and as a course for people who will major in other disciplines that require significant involvement with computing.

Materials

- Pencil and/or black/blue pen
- Binder
- Lined Filler Paper
- Memory Stick

Software Resources (all freely available online)

- Java JDK - java.oracle.com
- Greenfoot - www.greenfoot.org
- Eclipse - www.eclipse.org
- Codingbat code practice: http://codingbat.com/iava.
- https://repl.it/
- Microsoft Office 2016

Text & Print Resources

- Java Software Solutions for AP Computer Science A, third edition, John Lewis, William Loftus, Cara Cocking, 2011
- Barron's AP Computer Science A, 8th Edition, 2018
- Blue Pelican Java, Virtual Book Worm, 2005-2015

Grading Policy

Major Assessments	45%
Minor Assessments	20%
Daily	15%
Final Exam	20%

45% Major Assessments: Students will take likely have 6 or more tests each semester. Students should ensure they are fully prepared to take the test since no retakes are allowed. Students are strongly encouraged to be present on test days. When a student misses class the student is required to take the assessment on the scheduled day or on day of return if previously announced. Spring semester, students will be required to take a mock exam (Feb/Mar) and a final comprehensive exam (April). The length of these tests is such that the student may need to stay after school; this test cannot be completed during a regular classroom period.

20% Minor Assessments: Quizzes/Programs will be used throughout a unit to determine mastery of concepts along the way. This will serve as formative feedback and great study tools for unit test.

15% Daily Work and Homework: Homework is a vital part of learning and is a requirement for this class. It will be checked randomly. To receive full credit for homework, all parts of the assignment are expected to be attempted. When work is required, answers alone will not be sufficient enough to receive full credit. Missed homework may be completed for full credit during academic tutoring sessions.

20% Final Exam: Spring semester, students will take a final project encompassing topics learned.

Computer Language
This course will be taught using the Java programming language. Java is a good language for new programmers and is required for the AP Computer Science Examination. The AP Examination requires all coding to be in Java or no credit will be awarded.

Equipment
Students have access to computers during class as well as before and after school. These computers contain the Sun Java JDK as well as the Greenfoot and Eclipse development environments, the software we will be using. All software is freely available online.

Students may find it useful to be able to practice Java programming on their computers at home. All the software we use can be downloaded from the Internet free of charge.

Classroom Rules, Procedures and Student Expectations

Be Prepared	• Keep up with Classwork/Homework Keep up with your flash drive • Writing Utensil (pencil on assessment days) 3-ring binder/paper
Expect Excellence	• Participate in subject related activities for the entire class • period • All work turned in by a student must be work done by that student • Take notes • Keep an organized notebook • Don't get behind. It is a mistake to leave your reading and homework until the last minute
Always Be on Time	• Be on time to class each day • Turn in work on time • Make up work on time
Respect Everyone	• Respect the opinions of others • Respect your teacher and her decisions • Do not make disrespectful comments about other students, teachers, or school staff • Be able to work both individually and cooperatively • Remember that cooperative work requires an equal amount • of contribution from all members of the group
Show Bear Pride	• Do not bring outside food/drink into class (exception: bottled water in a sealable container) • Keep the classroom clean and throw away trash • Always have a positive attitude in class! • Pay attention to due dates • Plan for your future • Take responsibility for your actions • Read and follow all rules in the Student Handbook

Tutoring: Tutoring will be available most days after school and in the morning by appointment. Tutoring will also be available during academic opportunity periods (Bear Time). Tutoring is highly recommended for struggling students.

Consequences for Failure to Follow Rules and Procedures:

Step 1: Warning
Step 2: Warning & Parent Email
Step 3: Parent phone call and/or detention
Step 4: Office referral

Please Note: The discipline plan is progressive and follows the PBIS structure. Some offenses may automatically move a student to step 4 (for example: severe disrespect, fighting, cheating, etc.)

Cheating: Any student caught cheating will be referred to the office under a discipline referral. Refer to the handbook for consequences. In computer science cheating includes using major portions someone else's program as your own and failing to credit someone who assisted you with a program. Remember that Codingbat assignments are included in this policy. Refer to the handbook for consequences.

Late/Missing Work Procedure: Assignments not turned in on time will result in academic detention. Failure to serve academic detention will result in an office referral. Refer to the handbook for the late work procedures.

Teacher Communication: I answer email, as well as return phone messages, and encourage parents to contact me regarding the education of their student. I encourage you and your student to monitor the grades in this course. Please don't hesitate to contact me with questions or concerns.

Occasionally email does not go through. If you do not hear from me a day or two after sending an email, assume I did not receive the email and pursue a different form of communication.

Topic Outline

Following is an outline of the major topics covered by the AP Examination in Computer Science A.

I. Object-Oriented Program Design
The overall goal for designing a piece of software (a computer program) is to correctly solve the given problem. At the same time, this goal should encompass specifying and designing a program that is understandable, can be adapted to changing circumstances, and has the potential to be reused in whole or in part. The design process needs to be based on a thorough understanding of the problem to be solved.

A. Program design
1. Read and understand a problem's description, purpose, and goals.
Apply data abstraction and encapsulation.
2. Read and understand class specifications and relationships among the classes ("is-a", "has-a" relationships).
Understand and implement a given class hierarchy.
3. Identify reusable components from existing code using classes and class libraries.

B. Class design
1. Design and implement a class.
2. Choose appropriate data representation and algorithms.
Apply functional decomposition.
3. Extend a given class using inheritance.

II. Program Implementation
The overall goals of program implementation parallel those of program design. Classes that fill common needs should be built so that they can be reused easily in other programs. Object-oriented design is an important part of program implementation.

A. Implementation techniques
1. Methodology
a. Object-oriented development
b. Top-down development
c. Encapsulation and information hiding
d. Procedural abstraction

B. Programming constructs
   1. Primitive types vs. objects
   2. Declaration
      a. Constant declarations
      b. Variable declarations
      C. Class declarations
      d. Interface declarations
      e. Method declarations
      f. Parameter declarations
   3. Console output(System.out.print/println)
   4. Control
      a. Methods
      b. Sequential
      C. Conditional
      d. Iteration
      e. Understand and evaluate recursive methods
   5. Java library classes (included in the AP Java Subset)

Ill. Program Analysis

The analysis of programs includes examining and testing programs to determine whether they correctly meet their specifications. It also includes the analysis of programs or algorithms in order to understand their time and space requirements when applied to different data sets.

A. Testing
   1. Test classes and libraries in isolation
   2. Identify boundary cases and generate appropriate test data
   3. Perform integration testing
B. Debugging
   1. Categorize errors: compile-time, run-time 1 logic
   2. Identify and correct errors
   3. Techniques: use a debugger, add extra output statements, hand-trace code
C. Understand and modify existing code
D. Extend existing code using inheritance
E. Understand error handling
   1. Understand runtime exceptions
F. Reason about programs
   1. Pre-and post-conditions
   2. Assertions
G. Analysis of algorithms
   1. Informal comparisons of running times
   2. Exact calculation of statement execution counts
H. Numerical representations and limits
   1. Representations of numbers in different bases
   2. Limitations of finite representations (e.g., integer bounds, imprecision of floating-point representations, and round-off error

IV. Standard Data Structures

Data structures are used to represent information within a program. Abstraction is an important theme in the development and application of data structures.

A. Simple data types (int, boolean, double)
B. Classes
C. Arrays

V. Standard algorithms

Standard algorithms can serve as examples of good solutions to standard problems. Programs implementing them can serve as models of good program design. They provide examples for analysis of program efficiency. Many are intertwined with standard data structures.

A. Operations on data structures listed above
   1. Traversals
   2. Insertion
   3. Deletion
B. Searching
   1. Sequential
   2. Binary
C. Sorting
   1. Selection
   2. Insertion
   3. Mergesort

VI. Computing in Context

A working knowledge of the major hardware and software components of computer systems is necessary for the study of computer science, as is the awareness of the ethical and social implications of computing systems. These topics need not be covered in detail, but should be considered throughout the course.

A. System reliability
B. Privacy
C. Legal issues and intellectual property
D. Social and ethical ramifications of computer use