Course Overview
2017 - 2018

AP Computer Science A

course webpage: tiny.cc/WoodstockCS

Woodstock Union High School
Andrew Smith

1. Objects & Randomness	2. Inheritance & Arrays	3. Abstract Classes & Lists	4. Strings & Basic Recursion	5. Advanced Recursion	6. Searching & Sorting	7. AP Exam Prep	8. Final Project
How can I roll a hundred dice at the same time, but only write the code once?	How do I work with lists of things?	How can I build the Asteroids game?	What happens when a function calls itself?	What happens if I zoom in forever?	How does Google Search work so fast?	How can I do my best on the AP exam?	How can I use software to improve someone's life?
Aug 30 - Sep 29 20.5 days	Oct 2 - Nov 1 20 days	Nov 2 - Dec 7 19.5 days	Dec 8 - Jan 12 17.5 days	Jan 22 - Feb 16 18 days	Feb 26 - Mar 30 19 days	Apr 2 - May 8 19.5 days	May 9 - June 1 16 days
Standards **S00**: listens to understand and asks genuine questions **S05**: makes a meaningful contribution to the community **S30**: Classes **S31**: Basic control **S40**: Iteration **S41**: Variables **S42**: Java Subset **S43**: Processing	Standards **S01**: purposeful private reasoning time and explain my reasoning **S05**: makes a meaningful contribution to the community **S20**: Problem analysis **S21**: Class and interface specifications, relationships, and inheritance **S32**: Data abstraction and encapsulation **S35**: Interfaces **S36**: Primitives **S38**: Arrays (1+2 dim.) **S45**: Methods and parameters **S47**: Numeric expressions	Standards **S02**: compare ideas and multiple pathways **S05**: makes a meaningful contribution to the community **S06**: Social and ethical ramifications of computer use **S33**: Code reuse **S34**: Data representation and algorithms **S37**: Lists **S43**: Implementation techniques **S49**: Strings **S51**: Debugging **S52**: Runtime exceptions	Standards **S03** critique and debate and using reasoning as the authority **S05**: makes a meaningful contribution to the community **S10**: presents work to a lay audience **S44**: Primitive types vs. reference types **S46**: Recursion **S49**: Strings **S50**: Testing **S52**: Runtime exceptions **S53**: Algorithm Analysis	Standards **S00**: listens to understand and asks genuine questions **S05**: makes a meaningful contribution to the community **S20**: Problem analysis **S21**: Class and interface specifications, relationships, and inheritance **S35**: Interfaces **S38**: Arrays (1+2 dim.) **S44**: Primitive types vs. reference types **S46**: Recursion **S48**: Boolean expressions, short-circuit evaluation, De Morgan's law	Standards **S01**: purposeful private reasoning time and explain my reasoning **S05**: makes a meaningful contribution to the community **S22**: Functional decomposition **S23**: Searching **S24**: Sorting **S37**: Lists **S41**: Variables **S42**: Java library classes and interfaces included in the AP Java Subset	Standards **S02**: compare our ideas and explore multiple pathways **S05**: makes a meaningful contribution to the community **S06**: Social and ethical ramifications of computer use **S23**: Searching **S24**: Sorting **S30**: Classes **S38**: Arrays (1+2 dim.) **S50**: Testing **S54**: Representations of nonnegative integers in different bases **S55**: Implications of finite integer bounds	Standards **S03**: critique and debate and using reasoning as the authority **S05**: makes a meaningful contribution to the community **S06**: Social and ethical ramifications of computer use **S10**: presents work to a lay audience S20: Problem analysis **S25**: System reliability **S50**: Testing
Strands **P0**: collaborating and fostering inclusive culture **P3**: developing and using abstractions **P4**: creating computational artifacts	Strands: **P0**: collaborating and fostering inclusive culture **P2**: recognizing and defining computational problems **P3**: developing and using abstractions **P4**: creating computational artifacts	Strands **P0**: collaborating and fostering inclusive culture **P3**: developing and using abstractions **P4**: creating computational artifacts **P5**: testing and refining computational artifacts	Strands **P0**: collaborating and fostering inclusive culture **P1**: communicating about computing **P4**: creating computational artifacts **P5**: testing and refining computational artifacts	Strands **P0**: collaborating and fostering inclusive culture **P2**: recognizing and defining computational problems **P3**: developing and using abstractions **P4**: creating computational artifacts	Strands **P0**: collaborating and fostering inclusive culture **P2**: recognizing and defining computational problems **P3**: developing and using abstractions **P4**: creating computational artifacts	Strands **P0**: collaborating and fostering inclusive culture **P2**: recognizing and defining computational problems **P3**: developing and using abstractions **P5**: testing and refining computational artifacts	Strands **P0**: collaborating and fostering inclusive culture **P1**: communicating about computing **P2**: recognizing and defining computational problems **P5**: testing and refining computational artifacts