

Advanced Placement (AP) Computer Science

Course Overview

This course exposes students to computer programming in the language java, utilizing the java compiling software: “BlueJ.” The material covered in this course is equivalent to that seen in a one-semester introductory college level computer science course. Students will be expected to use logical reasoning and problem solving skills to create and analyze a variety of computer programs. (Prerequisites: Algebra II)

Units of Study	Essential Questions	Topics	Instructional Strategies
Introduction to Programming	<ul style="list-style-type: none"> How do variables and operations in programming differ from variables and operations in mathematics? 	<ul style="list-style-type: none"> Variable Declaration/Instantiation Operations Interactive Programming Object Oriented Programming (1) 	<ul style="list-style-type: none"> Exemplar Presentations are given to students to assist with program writing. These are programs that complete a similar task, to use as scaffolding for completing a similar task. Students are encouraged to pair up to complete in-class tasks. Computer Programming is learned best by bouncing ideas off of each other, and filling in each other’s missing logic. Practice Exams and Exam excerpts are given frequently which are obviously a great mechanism to prepare for the exam itself, and establish benchmarks for where students need improvement. 3 “case studies” are examined to give students a taste of how the beginning programming skills that they are learning apply to real-life complete programs which complete impressive tasks. Some topics are best
Control of Flow	<ul style="list-style-type: none"> What advances do “if statements” and “loops” provide the programmer? What are differences between “while loops” and “for loops?” 	<ul style="list-style-type: none"> Boolean expressions “if statements” “while loops” “for loops” Iterators 	
Class and Method structure	<ul style="list-style-type: none"> How can the Class structure in programming be related to objects and actions in real life? Why is it important to include that conventional methods in every class created? 	<ul style="list-style-type: none"> Object Oriented Programming (2) Class Structure Method Structure Conventional Methods Overloading (1) 	
Class Enhancements	<ul style="list-style-type: none"> How the words “public,” “static,” and “private” define themselves? How can knowledge of Exceptions help the programming when debugging? 	<ul style="list-style-type: none"> References Modifiers Exceptions and Debugging 	
Arrays	<ul style="list-style-type: none"> When is it more valuable to 	<ul style="list-style-type: none"> Indexing and 	

	<p>use the ArrayList Class, and when must the programmer use primitive array types?</p> <ul style="list-style-type: none"> • What reasons does the programmer have for starting indexing at zero? 	<p>Insert/Remove</p> <ul style="list-style-type: none"> • 2D Arrays • Searching • ArrayLists 	<p>introduced by “building” the programs from scratch as a class, and receiving student input where different avenues of coding are possible.</p>
Inheritance	<ul style="list-style-type: none"> • How does the concept of “inheritance” in programming relate to the animal kingdom in our world? • How does overriding methods differ from overloading methods? 	<ul style="list-style-type: none"> • Subclasses • Overriding • Class Hierarchies • Polymorphism • Interfaces 	
Recursive Programming	<ul style="list-style-type: none"> • When is recursive programming beneficial to a program? • Which sorting method, considering both efficiency and logistics, is the most practical to use? 	<ul style="list-style-type: none"> • Recursive Methods • Sorting 	