

# AP Mentoring Year-Long Session Outline

## AP Computer Science A

Session	Topics	Possible Objectives (to be selected and tailored based on mentee need)
1	Course Introductions	<ul style="list-style-type: none"> <li>▪ Become familiar with mentor and mentee peers.</li> <li>▪ Establish goals for the mentoring program.</li> <li>▪ Begin assessing own course and teaching.</li> <li>▪ Build a yearlong plan that provides opportunities to develop the enduring understandings of the course while preparing them for the end-of-course assessment. (B.1)</li> <li>▪ Prepare students for the variety of question types that will be used in the end-of-course assessment. (D.1)</li> <li>▪ If using an AP-endorsed provider, adopt their syllabus. Otherwise, adopt the College Board Unit Guides or use the AP Course Audit syllabus development resources to create and submit a syllabus for approval. (E.1)</li> <li>▪ Log in to myap.collegeboard.org and build class sections for their course. (E.2)</li> </ul>
2	Course Implementation	<ul style="list-style-type: none"> <li>▪ Create a classroom environment that provides a sense of belonging for all students. (A.1)</li> <li>▪ Become familiar with the specific Java classes and syntax that are in the CED and included on the end-of-year assessment. (A.2)</li> <li>▪ Select an AP-endorsed provider for their curriculum or select resources to create their own curriculum. (A.3)</li> <li>▪ Choose an IDE and instructional resources that best suit students, school, and teacher needs. (B.2)</li> <li>▪ Use an AP-endorsed provider or the Unit Guides in the CED to sequence and scaffold the skills and concepts in an appropriate way for their students. (B.3)</li> <li>▪ Understand the scope of the content and skills students will need to demonstrate understanding of on the end-of-year assessment. (E.3)</li> <li>▪ Engage in the AP Computer Science A Online Teacher Community for support, ideas, and resources. (E.4)</li> </ul>
3	Connecting Content and Skills	<ul style="list-style-type: none"> <li>▪ Select engaging and relevant course materials that will appeal to a diverse group of students. (A.4)</li> <li>▪ Use an AP-endorsed provider or the Unit Guides in the CED to sequence and scaffold the skills and concepts in an appropriate way for their students. (B.3)</li> <li>▪ Use examples, practice, and projects that encourage students to incorporate their own interests and connect computer science to other fields of study and the real world. (C.1)</li> </ul>

		<ul style="list-style-type: none"> <li>▪ Provide instructional approaches, strategies, and activities for students with varying levels of experience and understanding. (C.2)</li> <li>▪ Prepare students for content to be tested using a variety of different skills. (D.2)</li> <li>▪ Guide students in registering for the appropriate section in AP Classroom and assign assessments. (E.5)</li> <li>▪ Provide opportunities, especially early in the course, for students to evaluate individual lines of code to ensure that they can apply the meaning of Java operators included in the Course and Exam Description. (G.1)</li> <li>▪ Provide scaffolded opportunities for students to analyze program code containing conditional statements and iteration and using a variety of content to determine the result. (G.2)</li> <li>▪ Provide scaffolded opportunities for students to analyze program code containing method calls and using a variety of content to determine the result. (G.3)</li> <li>▪ Provide scaffolded opportunities for students to analyze program code containing iteration or recursion to determine</li> </ul>
4	Formative vs. Summative Assessments	<ul style="list-style-type: none"> <li>▪ Build a yearlong plan that provides opportunities to develop the enduring understandings of the course while preparing them for the end-of-course assessment. (B.1)</li> <li>▪ Use an AP-endorsed provider or the Unit Guides in the CED to sequence and scaffold the skills and concepts in an appropriate way for their students. (B.3)</li> <li>▪ Provide feedback to students on a variety of assignments and at multiple stages in a large project to improve student understanding and performance. (C.3)</li> <li>▪ Prepare students for content to be tested using a variety of different skills. (D.2)</li> <li>▪ Prepare students to answer free response questions by providing accurate feedback to students on administered practice free response questions based on the scoring guidelines. (D.3)</li> <li>▪ Guide students in registering for the appropriate section in AP Classroom and assign assessments. (E.5)</li> <li>▪ Use data from both formative questions and summative questions to inform instruction. (E.6)</li> </ul>
5	Labs and Programming Projects	<ul style="list-style-type: none"> <li>▪ Use examples, practice, and projects that encourage students to incorporate their own interests and connect computer science to other fields of study and the real world. (C.1)</li> <li>▪ Provide instructional approaches, strategies, and activities for students with varying levels of experience and understanding. (C.2)</li> <li>▪ Provide feedback to students on a variety of assignments and at multiple stages in a large project to improve student understanding and performance. (C.3)</li> <li>▪ Provide opportunities for students to engage in larger programming assignments or labs that utilize multiple existing classes. (C.4)</li> <li>▪ Provide scaffolded opportunities for students to write program code to create objects of a class and call methods. (H.1)</li> <li>▪ Provide scaffolded opportunities for students to write program code to define a new type by</li> </ul>

		<p>creating a class. (H.2)</p> <ul style="list-style-type: none"> <li>▪ Provide scaffolded opportunities for students to write program code to satisfy method specifications using expressions, conditional statements, and iterative statements. (H.3)</li> <li>▪ Provide scaffolded opportunities for students to write program code to create, traverse, and manipulate elements in 1D array or <code>ArrayList</code> objects. (H.4)</li> <li>▪ Provide scaffolded opportunities for students to write program code to create, traverse, and manipulate elements in 2D array objects. (H.5)</li> </ul>
6	Object Oriented Programming	<ul style="list-style-type: none"> <li>▪ Provide opportunities for students to design program code themselves, determine missing code needed so that the code segment functions, and interact with classes that they did not write, including classes in inheritance hierarchies. (F.1)</li> <li>▪ Provide scaffolded opportunities for students to analyze program code containing method calls and using a variety of content to determine the result. (G.3)</li> <li>▪ Provide scaffolded opportunities for students to analyze program code containing iteration or recursion to determine the number of times a code segment will execute. (G.4)</li> <li>▪ Provide scaffolded opportunities for students to write program code to create objects of a class and call methods. (H.1)</li> <li>▪ Provide scaffolded opportunities for students to write program code to define a new type by creating a class. (H.2)</li> <li>▪ Provide scaffolded opportunities for students to identify errors and use test-cases to validate results and determine if two or more code segments yield equivalent results. (I.1)</li> <li>▪ Provide scaffolded opportunities for students to analyze program code in a general way, without using specific values. (J.1)</li> </ul>
7	Maintaining Motivation and Engagement	<ul style="list-style-type: none"> <li>▪ Use recruitment strategies to create a classroom that is representative of their school population in terms of gender and race. (A.5)</li> <li>▪ Develop instruction, activities, and projects that are just beyond the current ability level and understanding of students and that integrate student choice to make them more engaging. (B.4)</li> <li>▪ Use examples, practice, and projects that encourage students to incorporate their own interests and connect computer science to other fields of study and the real world. (C.1)</li> <li>▪ Provide instructional approaches, strategies, and activities for students with varying levels of experience and understanding. (C.2)</li> <li>▪ Provide opportunities for students to engage in larger programming assignments or labs that utilize multiple existing classes. (C.4)</li> </ul>
8	Data Structures	<ul style="list-style-type: none"> <li>▪ Use examples, practice, and projects that encourage students to incorporate their own interests and connect computer science to other fields of study and the real world. (C.1)</li> <li>▪ Provide opportunities for students to design program code themselves, determine missing code needed so that the code segment functions, and interact with classes that they did not write, including classes in inheritance hierarchies. (F.1)</li> <li>▪ Provide scaffolded opportunities for students to write program code to create, traverse, and</li> </ul>

		<p>manipulate elements in 1D array or <code>ArrayList</code> objects. (H.4)</p> <ul style="list-style-type: none"> <li>Provide scaffolded opportunities for students to write program code to create, traverse, and manipulate elements in 2D array objects. (H.5)</li> </ul>
9	Preparing for the Exam	<ul style="list-style-type: none"> <li>Prepare students for the variety of question types that will be used in the end-of-course assessment. (D.1)</li> <li>Prepare students for content to be tested using a variety of different skills. (D.2)</li> <li>Prepare students to answer free response questions by providing accurate feedback to students on administered practice free response questions based on the scoring guidelines. (D.3)</li> <li>Guide students in registering for the appropriate section in AP Classroom and assign assessments. (E.5)</li> <li>Use data from both formative questions and summative questions to inform instruction. (E.6)</li> <li>Provide scaffolded opportunities for students to identify errors and use test-cases to validate results and determine if two or more code segments yield equivalent results. (I.1)</li> <li>Provide scaffolded opportunities for students to analyze program code in a general way, without using specific values. (J.1)</li> </ul>
10	Professional and Course Development	<ul style="list-style-type: none"> <li>Reflect upon and review progress and growth</li> <li>Plan for continued growth of both the mentee and their course</li> </ul>