

# AP Computer Science A

## Scoring Materials for Digital Exam Practice

**Please note:** the digital exam practice resource was developed for students to complete technology checks, experience the digital platform, and practice answering exam questions, including each type of multiple-choice and free-response question they will encounter on exam day. Be aware that on exam day students will complete four free-response questions while this practice includes only two.

This digital exam practice is not a full-length exam, and it does not represent the complete scope of content and skills that students will see on the actual AP exam. This digital exam practice includes only content that would typically be taught in the first half of the school year, following the unit sequence in the AP Computer Science A Course and Exam Description. For more information on the 2021 Exam format, please visit: [apcentral.collegeboard.org/pdf/ap-2021-exam-formats.pdf](https://apcentral.collegeboard.org/pdf/ap-2021-exam-formats.pdf)

AP Exams are scored differently than traditional high school or college exams. When an AP Exam is administered, psychometric analysis determines the score ranges corresponding with each AP Exam score (5, 4, 3, 2, and 1) based on a composite score scale that combines and weights the different exam parts. Earning 40-50% of the available points can result in a score of 3 or better on many AP Exams. However, because the number of points corresponding with each AP Exam score can vary on different exams, students and teachers should not use the results of the digital exam practice to predict performance on the 2021 AP Exam.

## Multiple-Choice Answer Key

Multiple-Choice Question	Answer
1	E
2	C
3	A
4	D
5	D
6	A
7	A
8	D
9	B
10	D

## Applying the Scoring Criteria

Apply the question scoring criteria first, which always takes precedence. Penalty points can only be deducted in a part of the question that has earned credit via the question rubric. No part of a question (a, b, c) may have a negative point total. A given penalty can be assessed only once for a question, even if it occurs multiple times or in multiple parts of that question. A maximum of 3 penalty points may be assessed per question.

### 1-Point Penalty

- v) Array/collection access confusion (`[] get`)
- w) Extraneous code that causes side-effect (e.g., printing to output, incorrect precondition check)
- x) Local variables used but none declared
- y) Destruction of persistent data (e.g., changing value referenced by parameter)
- z) Void method or constructor that returns a value

### No Penalty

- Extraneous code with no side-effect (e.g., valid precondition check, no-op)
- Spelling/case discrepancies where there is no ambiguity\*
- Local variable not declared provided other variables are declared in some part
- `private` or `public` qualifier on a local variable
- Missing `public` qualifier on class or constructor header
- Keyword used as an identifier
- Common mathematical symbols used for operators (`*` `•` `÷` `≤` `≥` `<>` `≠`)
- `[]` vs. `()` vs. `<>`
- `=` instead of `==` and vice versa
- `length/size` confusion for array, `String`, `List`, or `ArrayList`; with or without `()`
- Extraneous `[]` when referencing entire array
- `[i,j]` instead of `[i][j]`
- Extraneous size in array declaration, e.g., `int[size] nums = new int[size];`
- Missing `;` where structure clearly conveys intent
- Missing `{ }` where indentation clearly conveys intent
- Missing `()` on parameter-less method or constructor invocations
- Missing `()` around `if` or `while` conditions

*\*Spelling and case discrepancies for identifiers fall under the “No Penalty” category only if the correction can be **unambiguously** inferred from context, for example, “Arraylist” instead of “ArrayList”. As a counterexample, note that if the code declares `int G=99, g=0;`, then uses `while (G < 10)` instead of `while (g < 10)`, the context does **not** allow for the reader to assume the use of the lower case variable.*

**Question 1: Methods and Control Structures****9 points****Canonical solution**

- (a)** `public boolean simulate()`  
{  
    int position = 0;  
  
    for (int count = 0; count < maxHops; count++)  
    {  
        position += hopDistance();  
  
        if (position >= goalDistance)  
        {  
            return true;  
        }  
        else if (position < 0)  
        {  
            return false;  
        }  
    }  
    return false;  
}
- (b)** `public double runSimulations(int num)`  
{  
    int countSuccess = 0;  
  
    for (int count = 0; count < num; count++)  
    {  
        if(simulate())  
        {  
            countSuccess++;  
        }  
    }  
    return (double)countSuccess / num;  
}

**5 points****4 points**

AP Computer Science A Digital Exam Practice Scoring Guidelines

(a) `simulate`

Scoring Criteria		Decision Rules	
1	Calls <code>hopDistance</code> and uses returned distance to adjust (or represent) the frog's position	Responses <b>can</b> still earn the point even if they use <code>hopDistance()</code> as a position, like <code>hopDistance() &lt; 0</code>  Responses <b>will not</b> earn the point if they only use <code>hopDistance()</code> as a count, like <code>hopDistance() &lt; maxHops</code>	1 point
2	Initializes and accumulates the frog's position at most <code>maxHops</code> times ( <i>must be in context of a loop</i> )	Responses <b>will not</b> earn the point if they do not use a loop	1 point
3	Determines if a distance representing multiple hops is at least <code>goalDistance</code>	Responses <b>can</b> still earn the point even if they use some number of hops * <code>hopDistance()</code> as the frog's final position	1 point
4	Determines if a distance representing multiple hops is less than starting position		1 point
5	Returns <code>true</code> if goal ever reached, <code>false</code> if goal never reached or position ever less than starting position	Responses <b>can</b> still earn the point if they have checks for all three conditions and correct return logic based on those checks, even if a check did not earn a point  Responses <b>will not</b> earn the point if they <ul style="list-style-type: none"> <li>do not check all three conditions</li> <li>only check for <code>goalDistance</code> after the loop</li> <li>only check for starting position after the loop</li> </ul>	1 point
<b>Total for part (a)</b>			<b>5 points</b>

AP Computer Science A Digital Exam Practice Scoring Guidelines

(b) `runSimulations`

Scoring Criteria		Decision Rules	
6	Calls <code>simulate</code> the specified number of times ( <i>no bounds errors</i> )	Responses <b>can</b> still earn the point even if they do not use the result of calling <code>simulate</code>	<b>1 point</b>
		Responses <b>will not</b> earn the point if they do not use a loop	
7	Initializes and accumulates a count of <code>true</code> results	Responses <b>will not</b> earn the point if they <ul style="list-style-type: none"> <li>initialize the count inside a loop</li> <li>do not use a loop</li> </ul>	<b>1 point</b>
8	Calculates proportion of successful simulations using <code>double</code> arithmetic	Responses <b>can</b> still earn the point if they perform the correct calculation on an accumulated value, even if there was an error in the accumulation	<b>1 point</b>
		Responses <b>will not</b> earn the point if they do not divide by the parameter	
9	Returns calculated value	Responses <b>will not</b> earn the point if they <ul style="list-style-type: none"> <li>calculate values using non-numeric types</li> <li>return a count of simulations</li> </ul>	<b>1 point</b>
			<b>Total for part (b) 4 points</b>
<b>Question-specific penalties</b>			
None			
			<b>Total for question 1 9 points</b>

## Applying the Scoring Criteria

Apply the question scoring criteria first, which always takes precedence. Penalty points can only be deducted in a part of the question that has earned credit via the question rubric. No part of a question (a, b, c) may have a negative point total. A given penalty can be assessed only once for a question, even if it occurs multiple times or in multiple parts of that question. A maximum of 3 penalty points may be assessed per question.

### 1-Point Penalty

- v) Array/collection access confusion (`[] get`)
- w) Extraneous code that causes side-effect (e.g., printing to output, incorrect precondition check)
- x) Local variables used but none declared
- y) Destruction of persistent data (e.g., changing value referenced by parameter)
- z) Void method or constructor that returns a value

### No Penalty

- Extraneous code with no side-effect (e.g., valid precondition check, no-op)
- Spelling/case discrepancies where there is no ambiguity\*
- Local variable not declared provided other variables are declared in some part
- `private` or `public` qualifier on a local variable
- Missing `public` qualifier on class or constructor header
- Keyword used as an identifier
- Common mathematical symbols used for operators (`*` `•` `÷` `≤` `≥` `<>` `≠`)
- `[]` vs. `()` vs. `<>`
- `=` instead of `==` and vice versa
- `length/size` confusion for array, `String`, `List`, or `ArrayList`; with or without `()`
- Extraneous `[]` when referencing entire array
- `[i,j]` instead of `[i][j]`
- Extraneous size in array declaration, e.g., `int[size] nums = new int[size];`
- Missing `;` where structure clearly conveys intent
- Missing `{ }` where indentation clearly conveys intent
- Missing `()` on parameter-less method or constructor invocations
- Missing `()` around `if` or `while` conditions

*\*Spelling and case discrepancies for identifiers fall under the “No Penalty” category only if the correction can be **unambiguously** inferred from context, for example, “Arraylist” instead of “ArrayList”. As a counterexample, note that if the code declares `int G=99, g=0;`, then uses `while (G < 10)` instead of `while (g < 10)`, the context does **not** allow for the reader to assume the use of the lower case variable.*

**Question 2: Class Design****9 points****Canonical solution**

```
public class StepTracker
{
    private int minSteps;
    private int totalSteps;
    private int numDays;
    private int numActiveDays;

    public StepTracker(int threshold)
    {
        minSteps = threshold;
        totalSteps = 0;
        numDays = 0;
        numActiveDays = 0;
    }

    public void addDailySteps(int steps)
    {
        totalSteps += steps;
        numDays++;
        if (steps >= minSteps)
        {
            numActiveDays++;
        }
    }

    public int activeDays()
    {
        return numActiveDays;
    }

    public double averageSteps()
    {
        if (numDays == 0)
        {
            return 0.0;
        }
        else
        {
            return (double) totalSteps / numDays;
        }
    }
}
```

**9 points**



# AP Computer Science A Digital Exam Practice Scoring Guidelines

## StepTracker

Scoring Criteria		Decision Rules	
<b>1</b>	Declares all appropriate <code>private</code> instance variables	Responses <b>will not</b> earn the point if they <ul style="list-style-type: none"> <li>omit keyword <code>private</code></li> <li>declare variables outside the class</li> </ul>	<b>1 point</b>
<b>2</b>	Declares constructor: <code>public StepTracker (int ____)</code>	Responses <b>can</b> still earn the point even if they omit keyword <code>public</code>  Responses <b>will not</b> earn the point if they declare the constructor <code>private</code>	<b>1 point</b>
<b>3</b>	Uses parameter and appropriate values to initialize instance variables	Responses <b>can</b> still earn the point even if they initialize primitive instance variables to default values when declared  Responses <b>will not</b> earn the point if they <ul style="list-style-type: none"> <li>do not use the parameter to initialize some instance variable</li> <li>do not declare instance variables</li> <li>initialize local variables instead of instance variables</li> <li>assign variables to parameters</li> </ul>	<b>1 point</b>
<b>4</b>	Declares header: <code>public void addDailySteps (int ____)</code>	Responses <b>can</b> still earn the point even if they omit keyword <code>public</code>  Responses <b>will not</b> earn the point if they declare the method <code>private</code>	<b>1 point</b>
<b>5</b>	Identifies active days and increments count in <code>addDailySteps</code> method	Responses <b>can</b> still earn the point even if they put valid comparison erroneously in some other method  Responses <b>will not</b> earn the point if they <ul style="list-style-type: none"> <li>do not use the parameter as part of the comparison</li> <li>do not increment a count of active days</li> <li>do not increment an instance variable</li> <li>compare parameter to some numeric constant</li> </ul>	<b>1 point</b>
<b>6</b>	Updates other instance variables appropriately in <code>addDailySteps</code> method	Responses <b>will not</b> earn the point if they <ul style="list-style-type: none"> <li>update another instance variable only on active days</li> <li>update another instance variable inappropriately</li> <li>do not update appropriate instance variable</li> <li>update a local variable</li> </ul>	<b>1 point</b>

AP Computer Science A Digital Exam Practice Scoring Guidelines

7	Declares and implements <code>public int activeDays()</code>	Responses <b>can</b> still earn the point if they return appropriate count of active days even if the instance variables were updated improperly in <code>addDailySteps</code> or <code>activeDays</code>	<b>1 point</b>
		Responses <b>will not</b> earn the point if they <ul style="list-style-type: none"> <li>• declare method <code>private</code></li> <li>• return value that is not the number of active days</li> <li>• do not return a value</li> </ul>	
8	Declares header: <code>public double averageSteps()</code>	Responses <b>can</b> still earn the point even if they omit keyword <code>public</code>	<b>1 point</b>
		Responses <b>will not</b> earn the point if they declare the method <code>private</code>	
9	Returns calculated number of steps <code>double average</code>	Responses <b>can</b> still earn the point even if they <ul style="list-style-type: none"> <li>• maintain instance variables improperly but calculate appropriate average</li> <li>• do not handle the special case where no days are tracked</li> </ul>	<b>1 point</b>
		Responses <b>will not</b> earn the point if they <ul style="list-style-type: none"> <li>• use integer division</li> <li>• calculate something other than steps divided by days</li> <li>• do not return a value</li> </ul>	
<b>Question-specific penalties</b>			
None			
<b>Total for question 2</b>			<b>9 points</b>