

*Ninth Edition*

Be Prepared  
for the  
**AP**  
Computer  
Science  
Exam in Java

**Maria Litvin**  
**Gary Litvin**

Skylight Publishing  
Andover, Massachusetts

**Copyright © 2026 by  
Maria Litvin, Gary Litvin, and Skylight Publishing**

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the authors and Skylight Publishing.

ISBN 978-0-9972528-3-5

Skylight Publishing  
9 Bartlet Street, Suite 70  
Andover, MA 01810

web: [www.skylit.com](http://www.skylit.com)  
email: [sales@skylit.com](mailto:sales@skylit.com)  
[support@skylit.com](mailto:support@skylit.com)

1 2 3 4 5 6 7      30 29 28 27 26

Printed in the United States of America

# Chapter 1 Exam Format, Grading, and Tips

## 1.1 Exam Format and Materials

Figure 1-1 shows the format of the AP Computer Science A exam. The exam takes three hours of test time, plus a break and time for instructions. The exam is divided into two sections. Section I consists of 42 multiple-choice questions with a total allotted time of 1 hour and 30 minutes (a little over two minutes per question, on average). Section II consists of four free-response questions with a total allotted time of 1 hour and 30 minutes. The first free-response question has two parts; the remaining three questions have one part each.

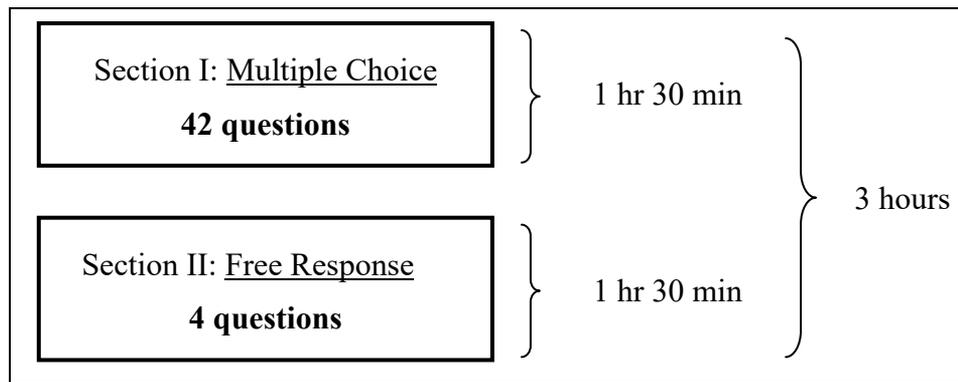


Figure 1-1. AP Computer Science A exam format

**The exam is administered online via the Bluebook test-taking app; scratch paper is provided.**

Bluebook will notify you five minutes before the time expires, both in the multiple-choice and free-response sections.

During the entire exam you will have access to the *Java Quick Reference*, which provides a list of the Java library classes and their methods included in the AP Java subset. Students and teachers can find a copy of the *Java Quick Reference* in the Course and Exam Description (CED). It is expected that you will already be very familiar and comfortable with the required Java library classes and their methods before the exam.

The multiple-choice section is a mixture of questions related to general computer science terms, program design decisions, specific elements of Java syntax, analysis of fragments of Java code, basic properties of a class, creating objects and accessing their methods. The multiple-choice section is graded out of 42 points with all questions weighted equally and no penalty for wrong answers. It accounts for 55% of the total exam score. The free-response section is graded out of 25 points and accounts for 45% of the total exam score.

The topics of the four free-response questions test specific skills:

Question 1: “Methods and Control Structures” (7 points total)

Part (a): Write a constructor or a method that may involve `if-else` and loops (4 points)

Part (b): Call `String` methods; you may need to call the method from Part (a) (3 points)

Question 2: “Class Design” (7 points)

Write a short class with instance variables, a constructor, and one method

Question 3: “Data Analysis with `ArrayList`” (5 points)

Write one method that examines and/or manipulates the elements of an `ArrayList`

Question 4: “2D Array” (6 points)

Write one method that examines and/or manipulates the elements of a 2D array

Part (b) of Question 1 may refer to the method implemented in Part (a). Each part is graded separately, and your implementation of Part (a) does not have to be correct in order for you to receive full credit for Part (b).

Your implementation of the class in Question 2 will be graded based on the correctness of the implementation of your class, including the *encapsulation* principle (declaring all instance variables `private`), and other criteria.

The College Board requires a typical AP CSA course to include at least 20 hours of computer programming lab work. The Development Committee has made available six sample labs for practice (available exclusively in the College Board’s AP Classroom).

**These labs are only examples designed to illustrate the scope and difficulty of lab work in a typical AP CSA course. The content and code in these labs will not be tested on the AP CSA exams.**

## 1.2 The AP Java Subset

There is no formal definition of the AP Java subset in the CED, but you can glean the elements of the subset from the *Essential Knowledge* components and the “Exclusion Statements” in the CED.

So what’s in the subset? Actually, quite a bit:

- Comments: `/* ... */`, `//`, and `/** ... */`; preconditions and postconditions.
- `boolean`, `int`, and `double` primitive data types. `(int)` and `(double)` casts. **Other primitive data types, including `char`, are not in the subset and should be avoided on the exam.**
- `Integer.MIN_VALUE` and `Integer.MAX_VALUE`
- The assignment operator `=`. The arithmetic operators `+`, `-`, `*`, `/`, `%`. The increment/decrement operators `++` and `--` (**use only the postfix form `k++` or `k--`, and do not use them in expressions**). The compound assignment operators `+=`, `-=`, `*=`, `/=`, `%=`. The relational operators `<`, `>`, `<=`, `>=`, `==`, `!=`. The logical operators `&&`, `||`, `!`.
- `if-else` statements. The keywords `true` and `false`.
- Literal strings in double quotes. `\n`, `\\`, and `\"` escape characters in literal strings. The `+` and `+=` operators for concatenating strings. `String`’s `length`, `substring` methods, `indexOf(String s)`, `compareTo`, `equals`, and `split` methods.
- `Integer.parseInt(String s)` and `Double.parseDouble(String s)` methods.
- `System.out.print` and `System.out.println`.
- `File(String str)` constructor. `Scanner(File f)` constructor. `IOException`. `Scanner`’s `nextInt`, `nextDouble`, `nextBoolean`, `nextLine`, `next`, `hasNext`, and `close` methods for reading a text file.
- One-dimensional and two-dimensional (**rectangular only**) arrays. `array.length`. Arrays of objects. Initialized arrays such as `int[] arr = {1, 2, 3};`  
`int[][] m = {{1, 2, 3}, {4, 5, 6}};`  
Default values of elements for arrays defined using the `new` operator.

- for and while loops. The “enhanced” for loop (for-each loop):  
for (*type* x : values)...
- Classes. Instance variables. Constructors. The new operator. public and private methods. static methods (may be public). Overloaded methods. The return statement. void methods.
- static variables and static final variables (constants). null. this reference. **All instance variables are private.**
- Understanding ArithmeticException, NullPointerException, IndexOutOfBoundsException, ArrayIndexOutOfBoundsException, ConcurrentModificationException, and InputMismatchException, and IOException.
- The ArrayList<E> class (see Section 2.6).
- File and Scanner classes (see Section 2.9).
- Other library classes, methods, and constants:
  - String:** length(), substring(int from), substring(int from, int to), indexOf(String s), equals, compareTo, and split
  - Integer:** Integer.parseInt(); Integer.MIN\_VALUE and Integer.MAX\_VALUE
  - Double:** Double.parseDouble()
  - Math:** abs(int x), abs(double x), pow(double base, double exp), sqrt(double x), random()
  - Object:** equals(Object other), toString()

Also understand the toString and equals methods for all objects and the compareTo method for the String, Integer, and Double objects.

**If you feel you must stray from this subset in your free-response solution, you might have misunderstood the problem and be making it harder than it is.**

At the same time, it is okay to use such out-of-the-subset features as Math.min(x, y) and Math.max(x, y) methods, ArrayList’s contains and remove(object) methods, break in loops, and other simple tools that all exam readers are familiar with.

Things that are **not** in the AP Java subset and should be avoided include the following:

- Java syntax abominations, such as the `?_:_` operator
- `++` and `--` in expressions (as in `a[i++]`) and the prefix form of the `++` and `--` operators (`++k`, `--k`)
- Primitive data types other than `int`, `double`, and `boolean` (`char`, `long`, and `float` are **not** in the subset)
- Bitwise logical operators and shift operators: `~`, `&`, `|`, `^`, `<<`, `>>`.
- The `switch` statement, the `do-while` loop, `break` and `continue` in loops
- Library classes and methods other than those mentioned in *Java Quick Reference*
- Abstract classes and interfaces
- Checked exceptions and `try-catch-finally` statements
- Any input other than `File`, `Scanner`, and `IOException` classes which are used only for reading from a text file. (Reading values from the keyboard using `Scanner` class is excluded).
- Any output other than `System.out.print` and `System.out.println`
- `enum` data types
- Subclasses and interfaces, `extends`, `implements`, `super`, method overriding

### 1.3 Tested Terms, Concepts, and Algorithms

In addition to Java, you need to be familiar with the following terms, concepts, and algorithms mentioned in the *Essential Knowledge* components of the CED:

- Rounding to the nearest integer: `(int) (x + 0.5)` for a positive double `x`; `(int) (x - 0.5)` for a negative `x`
- *Autoboxing / unboxing*
- Constructor and method *signature* (the method's name and the list of data types of parameters)
- *Invoking* constructors using `new`; calling methods using the “dot” operator
- *Local* variables; *instance variables* and *class (static) variables*
- Constructor and method *parameters*
- Passing arguments *by value* (an object is passed as a copy of the reference)
- *Accessor* and *mutator (modifier)* methods

- Array / ArrayList algorithms: traverse, find the minimum or maximum, find the sum or average of elements, examine all pairs of consecutive elements, find duplicates, examine all elements for a given property, shift or rotate elements left or right
- Sequential Search and iterative and recursive versions of Binary Search
- Selection Sort, Insertion Sort, and Mergesort
- Understand (**but not write**) recursive code

The Java subset’s features and the above terms, concepts, and algorithms are reviewed in the subsequent chapters.

## 1.4 Grading

The exam is graded on a scale from 1 to 5. Table 1-1 shows the College Board’s college credit recommendations and grade equivalents for AP grades.

| AP Grade | Credit Recommendation    | College Grade Equivalent |
|----------|--------------------------|--------------------------|
| 5        | Extremely well qualified | A                        |
| 4        | Well qualified           | A-, B+, B                |
| 3        | Qualified                | B-, C+, C                |
| 2        | Possibly qualified       | n/a                      |
| 1        | No recommendation        | n/a                      |

**Table 1-1. AP grades, credit recommendations, and college grade equivalents**

Grades of 5 and 4 are called “extremely well qualified” and “well qualified,” respectively, and usually will be honored by colleges that give credit or placement for AP CSA. A grade of 3, “qualified,” may be denied credit or placement at some colleges. Grades of 2 (“possibly qualified”) and 1 (“no recommendation”) will not result in college credit or placement.

Table 1-2 presents published statistics and grade distributions for the 2024 and 2023 exams. 98,136 students took the exam in 2024; 47.0 percent of them scored 4 or 5.

| AP Computer Science A | 2024   |       | 2023   |       |
|-----------------------|--------|-------|--------|-------|
|                       | Number | %     | Number | %     |
| Students              | 98,136 | 100.0 | 94,438 | 100.0 |
| Grade:                |        |       |        |       |
| 5                     | 25,137 | 25.6  | 25,269 | 26.8  |
| 4                     | 21,038 | 21.4  | 21,181 | 22.4  |
| 3                     | 19,754 | 20.1  | 17,730 | 18.8  |
| 2                     | 10,613 | 10.8  | 8,984  | 9.5   |
| 1                     | 21,594 | 22.0  | 21,274 | 22.5  |
| 4 or 5                | 46,175 | 47.0  | 46,450 | 49.2  |

**Table 1-2. 2024 and 2023 grade distributions**

**The multiple-choice section counts for 55% of the final grade, while the free-response section counts for 45%.**

The College Board uses a weighted combination of the multiple-choice (MC) and free-response (FR) scores to determine the final total score:

$$\text{totalScore} = \text{MC\_coeff} * \text{countCorrect} + \text{FR\_coeff} * \text{FR\_score};$$

One point is given for each correct answer to a multiple-choice question.

**There is no penalty for giving a wrong answer to a multiple-choice question, so it is a good strategy not to leave any answers blank. Cross out the choices that appear wrong and guess among the remaining ones. (Use Bluebook's cross-out feature.)**

Solutions to free-response questions are graded by a group of high school teachers and college professors. Scores are based on a *rubric* established by the Chief Reader, with input from the Exam Leader, Question Leaders, and Table Leaders. The total possible number of points is 25, with 7, 7, 5, and 6 points for Questions 1-4, respectively.

The final score is obtained by adding the MC and FR weighted scores. The MC and FR coefficients are chosen in such a way that they give 55% and 45% weights to the multiple-choice and free-response sections of the exam, respectively.

The cut-off points are determined by the Chief Reader in consultation with the College Board and may vary slightly from year to year based on the score distributions and close examination of a sample of individual exams. Table 1-3 shows the cut-off points from one of the past exams.

| Percent range | Grade |
|---------------|-------|
| 77.5 - 100    | 5     |
| 55.0 - 77.4   | 4     |
| 38.8 - 54.9   | 3     |
| 31.3 - 38.7   | 2     |
| 0 - 31.2      | 1     |

**Table 1-3. Sample score cut-off points for grades**

The College Board releases the free-response questions 48 hours after the exam and posts them on its website. We post our annotated solutions at [www.skylit.com/beprepared](http://www.skylit.com/beprepared) a few hours after the questions are posted. The College Board posts the scoring rubrics for the free-response questions sometime in the summer. Every few years the College Board releases a complete exam, including a diagnostic guide for the multiple-choice questions, scoring rubrics for the free-response questions, and the cut-off points for grades. See [www.skylit.com/beprepared/notes.txt](http://www.skylit.com/beprepared/notes.txt) for the latest updates.

## 1.5 College Credit

Most colleges will take your AP courses taken and exam grades into account in admission decisions. But acceptance of AP exam results for credit and/or placement varies widely among colleges. In general, the AP CSA course corresponds to a CS-1 course (Introductory Computer Science or Computer Programming I), a one-semester course for computer science majors. Some colleges base their decision on your grade, and some may not give any credit at all. Consult the websites of the colleges you are interested in and the College Board's AP credit policies page, <https://apstudent.collegeboard.org/creditandplacement/search-credit-policies>.

## 1.6 Exam-Taking Tips

Some things are obvious:

- If you took the time to read a multiple-choice question and all the answer choices, take an extra ten seconds and guess. Most likely you have eliminated one or two wrong answers even without noticing. **Do not leave any multiple-choice answers blank; there is no penalty for wrong answers.**
- Do read the question before jumping to the code included in the question. Notes to multiple-choice questions in our practice exams might teach you some shortcuts.

There are a few important things to know about answering free-response questions.

**In a nutshell: be neat, straightforward, and professional; keep your exam reader in mind, and don't show off.**

More specifically:

1. Stay within the AP Java subset, except for a few obvious shortcuts, such as `Math.max(...)` and `Math.min(...)`.
2. Remember that the elegance of your code does not count. More often than not, a brute-force approach is the best. If you write tricky, non-standard code, you may waste a lot of time and trick yourself in the process, or mislead your exam reader, who, after all, is only human. Your exam reader will read your solution but will not test it on a computer.
3. The efficiency of your code does not count, unless the desired performance of the solution is specifically stated in the question, which is unlikely.
4. Remember that Part (b) of Question 1 is graded independently from Part (a) and may actually be easier. It is not uncommon for the method specified in Part (a) to be called in Part (b). Do so, even if your Part (a) is incorrect or left blank. Do not re-implement code from Part (a) in Part (b) — you will waste valuable time and may lose points for doing so.
5. If a question presents a partial definition of a class with certain methods described but not implemented (“implementation not shown”), call these methods whenever appropriate in your code — do not write equivalent code yourself.

6. Bits of “good thinking” count. You may not know the whole solution, but if you have read and understood the question, go ahead and write fragments of code that may earn you partial credit. On the free-response Question 2, write the class header, constructor, and method headers even if you are not sure how to implement them. But don’t spend too much time improvising incorrect code.
7. You cannot submit two attempts at a solution in Bluebook. If you are unsure about your solution, copy and paste it at the top of the solution box, then review and edit as necessary. Once you are happier with the result, delete the previous version below. If you believe this attempt didn’t improve things, delete the copy at the top.
8. Read the comment above the method header quickly — it usually restates the task in a more formal way and sometimes gives valuable hints. Assume that all preconditions are satisfied — do not check them in your code!
9. One common mistake is to forget a `return` statement in a non-void method. Make sure the returned value matches the specified method return type.
10. Do not ignore hints in the question description. If an algorithm is suggested for a method (as in “you may use the following algorithm”), don’t fight it, just use it!
11. Remember that the exam readers score a vast number of exams in quick succession during a marathon grading session every June. Space out your code. Always indent your code properly. This helps you and your exam reader. If you miss a brace but your code is properly indented, the reader (as opposed to a Java compiler) may accept it as correct. Similarly, if you put each statement on a separate line, a forgotten semicolon might not be held against you.
12. Follow the Java naming convention: the names of all methods, variables, and parameters start with a lowercase letter. Use meaningful, but not too verbose, names for variables. `count` may be better than `a`; `sum` may be better than `temp`; `row`, `col` or `r`, `c` may be better than `i`, `j`. But `k` is better than `loopControlVariable`. If the question contains examples of code with names, use the same names when appropriate.

13. Don't bother with comments; they do not count, and you will lose valuable time. Occasionally you can write a very brief comment that indicates your intentions for the fragment of code that follows — but only if it helps you. For example:

```
// Find the first empty seat:  
...  
...
```

14. Don't worry about `imports` — assume that all the necessary Java library classes are imported.
15. Code strictly according to the specifications, preconditions, and postconditions. Avoid extraneous “bells and whistles” — you will lose points. Never add `System.out.print/println` in solutions unless specifically asked to do so. Do not read any data in your method — it is passed in as arguments or is available as instance or static variables of the class.
16. Do not use in your code specific numbers, strings, or dimensions of arrays given as examples in explanations of questions. If the question says, “For example, a two-dimensional array `pixelValues` may contain the following image” and shows an array of 4 rows by 5 columns, do not use 4 and 5 in your code — make your code work with an array of any size.
17. In Question 3, remember that you are working with an `ArrayList` and use its `size`, `get`, and `set` methods, not array's `length` and brackets for accessing elements.
18. In Question 4, given a 2D array `arr`, `arr.length` gives the number of rows and `arr[0].length` gives the number of columns. The first index is the row and the second index is the column, as in `arr[r][c]`.
19. Don't try to catch the exam authors on ambiguities: there will be no one to hear your case, and you'll waste your time. Instead, try to grasp quickly what was meant and write your solution.
20. Don't quit until the time is up. Use all the time you have and keep trying. The test will be over before you know it.