

Sixth Edition

Be Prepared
for the

AP

**Computer
Science
Exam in Java**

Maria Litvin

Phillips Academy, Andover, Massachusetts

Gary Litvin

Skylight Publishing, Andover, Massachusetts

Skylight Publishing
Andover, Massachusetts

**Copyright © 2014 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.

Library of Congress Control Number: 2013918747

ISBN 978-0-9824775-3-3

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

web: www.skylit.com
e-mail: sales@skylit.com
support@skylit.com

1 2 3 4 5 6 7 8 9 10 19 18 17 16 15 14

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 exam. The exam takes 3 hours of test time (plus breaks and time for instructions). It is divided into two sections. Section I consists of 40 multiple-choice questions with a total allotted time of 1 hour and 15 minutes (1.5 to 2 minutes per question on average). Section II consists of four free-response questions with a total allotted time of 1 hour and 45 minutes (about 25 minutes per question). The free-response questions usually consist of two or three parts each.

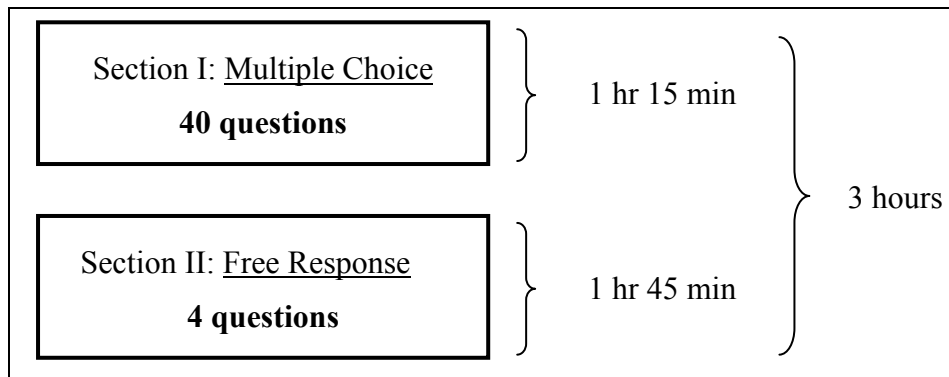


Figure 1-1. AP Computer Science exam format

No computers, calculators, other devices, books, or materials are allowed, only paper, pencil and an eraser.

Pens are allowed, but we recommend that you use a pencil.

At the exam you will receive the *Quick Reference* page, both in your multiple-choice questions booklet and again in your free-response questions booklet (in other words, the same *Quick Reference* page will be given to you twice). The *Quick Reference* provides a list of the Java library classes and their methods included in the AP subset. Teachers can obtain a copy of the *Quick Reference* at *AP Central* (see “A Guide to AP Central” at www.skylit.com). This document is provided for reference — it is expected that you will already be very familiar and comfortable with the required Java library classes 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, logical analysis of fragments of Java code, properties of classes, and OOP concepts.

The free-response section aims to cover a wide range of material: Java arithmetic, algorithms, one- and two-dimensional arrays, strings, classes and interfaces, Java library classes (within the AP subset), and so on. In past exams, students were asked to write a constructor or a method that performs a specified task under a given header for the method. The second part of the question often refers to the class or method implemented in the first part, but each part is graded separately, and your implementation of Part (a) does not have to be correct in order for you to get full credit for Part (b). Part (c) may ask questions about your implementation or ask you to write an additional method that uses Parts (a) and/or (b). In that case you are to assume that the methods in the previous parts work as intended, regardless of what you wrote for them.

Free-response questions may also include a “design” question, in which you are asked to design and write a complete small Java class. Your design and implementation 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.

In the past, the AP CS exams included a specific case study, developed by the Exam Development Committee. In 2008-2014, the case study was GridWorld.

Starting with the 2015 exam, neither GridWorld nor any other case study will be part of the exams.

The College Board expects a typical AP CS course to include at least 20 hours of computer lab work. The Development Committee has made available three sample labs.

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

1.2. The Java Subset

The Development Committee has defined a restricted subset of Java to be tested on the exam. The purpose of the subset is to focus the AP CS program more on general concepts than on the specifics of Java and to limit the scope, especially of material related to the peculiarities of Java. The subset is described in The College Board’s *Computer Science A Course Description*; we have a link to it from this book’s web site www.skylit.com/beprepared/.

What is in the subset? Actually, quite a bit:

- Comments: `/* ... */`, `//`, and `/** ... */`, `@param` and `@return` tags.
- `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.**
- Assignment (`=`), arithmetic (`+`, `-`, `*`, `/`, `%`), increment/decrement (`++`, `--`), compound assignment (`+=`, `-=`, `*=`, `/=`, `%=`), relational (`<`, `>`, `<=`, `>=`, `==`, `!=`), and logical (`&&`, `||`, `!`) operators. **Use only the postfix form of `++` and `--` (`k++` or `k--`), and do not use them in expressions.**
- `+` and `+=` operators for concatenating strings. `String`'s `compareTo`, `equals`, `length`, `substring`, and `indexOf(String s)` methods. `\n`, `\\`, and `\"` escape sequences in literal strings.
- `System.out.print` and `System.out.println`.
- One- and two-dimensional arrays, `array.length`, arrays of objects, initialized arrays such as `int[] x = {1, 2, 3};`
- `if-else`, `for`, including the “for each” form, `for(type x : values) ...`, `while`, `return`. **But `do-while` and `switch` are not included.**
- Classes. Constructors, the `new` operator, `public` and `private` methods, `static` methods, `static` variables and `static final` variables (constants), overloaded methods, `null`. **All instance variables are `private`.** Default initialization rules are not in the subset and won't come up on the exam.
- Inheritance, interfaces and abstract classes, `extends`, `implements`. Calling a superclass's constructor from a subclass (as in `super(...)`). Calling a superclass's method from a subclass (as in `super.someMethod(...)`). Passing `this` object to a method (as in `otherObject.someMethod(this)`).
- `NullPointerException`, `IndexOutOfBoundsException`, `ArrayIndexOutOfBoundsException`, `ArithmeticException`, `IllegalArgumentException`.

- Library classes, methods, and constants:

Object: equals(Object other), toString()
String: length(), substring(...), indexOf(String s)
Integer: Integer(int x), intValue();
 Integer.MIN_VALUE and Integer.MAX_VALUE
Double: Double(double x), doubleValue()
Math: abs(int x), abs(double x),
 pow(double base, double exp), sqrt(double x),
 random()

Also understand toString methods for all objects, the equals and compareTo methods for String, Integer, and Double. **Comparable<T> interface is not in the subset**, but is worth knowing.

- The List<E> interface and the ArrayList<E> class (see Section 2.6).

If you feel you must stray from the 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 OK to use such features outside the subset as Math.min(...) and Math.max(...) methods, ArrayList's contains and remove(object) methods, and other simple tools that all exam readers are familiar with.

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

- Java syntax abominations, such as the ?_:_ operator and the “comma” operator
- ++ and -- in expressions (as in a[i++])
- Primitive data types other than boolean, int, and double (char, long, float are not in the subset)
- Bitwise logical operators and shift operators: ~, &, |, ^, <<, >>.

Also not in the subset and will not be tested:

- The switch statement, the do-while loop, break and continue in loops
- The prefix form of ++ and -- operators (++k, --k)
- Library classes (such as StringBuffer, Arrays, DecimalFormat, etc.), unless specifically listed in the subset
- checked exceptions and try-catch-finally statements
- System.in and Scanner; any input and output other than System.out.print and System.out.println
- enum data types

1.3. Grading

The exam is graded on a scale from 1 to 5. 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 the AP exam in computer science. 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 get you college credit or placement.

Table 1-1 presents published statistics and grade distributions for the 2012 and 2013 exams. In 2013, 31,117 students took the exam; 53.2 percent scored 4 or 5.

AP Computer Science A	2013		2012	
	Number	%	Number	%
Students	31,117	100.0	26,103	100.0
Grade:				
5	8,268	26.6	6,171	23.6
4	8,276	26.6	6,351	24.3
3	4,339	13.9	4,075	15.6
2	2,168	7.0	2,013	7.7
1	8,066	25.9	7,493	28.7
4 or 5	16,544	53.2	12,522	48.0

Table 1-1. 2013 and 2012 grade distributions

The multiple-choice and free-response sections weigh equally in the final grade.

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.

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, Exam Leader, and Question Leaders. Each free-response question is graded out of 9 points, with partial credit given according to the rubric.

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 equal weights to the multiple-choice and free-response sections of the exam. For example, if the exam has 40 multiple-choice questions and 4 free-response questions, weights of 1.0 for multiple-choice and 1.1111 for free-response will give each section a maximum total of 40, for a maximum possible total score of 80.

Every year over the summer the College Board releases sample solutions to free-response questions and grading rubrics. Every few years the College Board releases a complete exam, including a diagnostic guide for multiple-choice questions, grading rubrics for free-response questions, and the cut-off points for grades.* Table 1-2 shows the cut-off points used for the 2004 and 2009 exams. The maximum possible composite score was 80 ($1.00 * MC + 1.1111 * FR$). In those years there was a 1/4 point penalty for incorrect answers to multiple-choice questions; there is no penalty now. The table shows both the actual cut-offs and the cut-offs adjusted for no-penalty grading. 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. In 2009, 77.5 percent or more would get you a 5.

AP Grade	2004		AP Grade	2009	
	Actual	Adjusted for no penalty		Actual	Adjusted for no penalty
5	63 - 80	65 - 80	5	60 - 80	62 - 80
4	49 - 62	52 - 64	4	44 - 59	47 - 61
3	39 - 48	43 - 51	3	33 - 43	37 - 46
2	32 - 38	36 - 42	2	25 - 32	29 - 36
1	0 - 31	0 - 35	1	0 - 24	0 - 28

Table 1-2. Score-to-grade conversion.

Statistical analysis of the 2009 exam shows that over 98.5 percent of students who answered correctly at least 27 out of 40 questions on the multiple-choice section received a 4 or a 5 for the whole exam.

* When this book went to print, 2009 was the last released exam. See www.skylit.com/beprepared for the latest updates.

1.4. College Credit

Most colleges will take your AP courses and exam grades into account in admission decisions if you take your exams before your senior year. But acceptance of AP exam results for credit and/or placement varies widely among colleges. In general, the AP Computer Science course corresponds to a CS-1 course (Introductory Computer Science or Computer Programming I), a one-semester course for computer science majors. Some colleges may base their decision on your grade, and some may not give any credit at all.

The AP program in computer science is a rigorous and demanding program that is comparable to or exceeds the level of the respective first-semester computer science courses at most colleges.

If you plan to major in computer science and your college of choice does not recognize a good grade on the AP exam for credit and/or placement, you should examine the reasons carefully. Decide for yourself whether these reasons are valid or just stem from the bias of that college or its computer science department.

1.5. 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.**
- If a common paragraph refers to a group of questions and you took the time to read it, try each question in the group.
- Do read the question before jumping to the code included in the question. Notes to multiple-choice questions in our practice exams might show you some shortcuts.
- In questions with I, II, III options, work from the answers; for example, you might be able to eliminate two or three answer choices if you are sure that Option I doesn't work.

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

Remember that all free-response questions have equal weight. Don't assume that the first question is the easiest and the last is the hardest.

In a nutshell: be neat, straightforward, and professional; keep your exam reader in mind; 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. You may waste a lot of time writing tricky, non-standard code 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. Superior efficiency of your code does not count, unless the desired performance of the solution is specifically stated in the question.
4. Remember that Parts (b) and (c) of a question are graded independently from the previous parts, and may actually be easier: Part (a) may ask you to write a method, while Part (b) or Part (c) may simply ask you to use it. It is common for method(s) specified in Part (a) to be called in subsequent parts. Do so, even if your Part (a) is incorrect or left blank. Do not re-implement code from earlier parts in later parts — 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 points. But don't spend too much time improvising incorrect code.
7. Don't waste your time erasing large portions of work. Instead, cross out your work neatly, but only after you have something better to replace it with. Do not cross out a solution if you have no time to redo it, even if you think it is wrong. You won't be penalized for incorrect code and may get partial credit for it. Exam readers are instructed not to read any code that you have crossed out. But if you wrote two solutions, be sure to cross one out: otherwise only the first one on the page will be graded.

8. Read the comment above the method header quickly — it usually restates the task in a more formal way and sometimes gives hints. Assume that all preconditions are satisfied — don't add unnecessary checks to 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 type.
10. Do not ignore any 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 grade a vast number of exams in quick succession during a marathon grading session every June. Write as neatly as possible. Space out your code (don't save paper).
12. 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 may not be held against you.
13. 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` 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.
14. Don't bother with comments; they do not count and you will lose valuable time. Occasionally you can put a very brief comment that indicates your intentions for the fragment of code that follows. For example:

```
    // Find the first empty seat:
    ...
    ...
```
15. Don't worry about `imports` — assume that all the necessary Java library classes are imported.
16. Code strictly according to the specifications and 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 a parameter.

17. 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.
18. 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 answer.
19. 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.

Other Computer Science and Mathematics Titles from Skylight Publishing

*Java Methods: Object-Oriented Programming and Data Structures,
Second AP Edition* ISBN 978-0-9824775-7-1

*250 Multiple-Choice Computer Science Questions in Java,
E-Edition* Part 978-0-9727055-J-E

*100 Multiple-Choice Questions in C++,
E-Edition* Part 978-0-9654853-C-E

*Mathematics for the Digital Age and Programming in Python,
Second Edition* ISBN 978-0-9824775-4-0

Be Prepared for the AP Calculus Exam, Second Edition
ISBN 978-0-9824775-5-7

800 Questions in Calculus with Solutions, E-Edition
Part 978-0-9727055-E-E

Calculus Calculator Labs Student Pack
Part 978-0-9727055-S-L

Calculus Calculator Labs Teacher Pack
Part 978-0-9727055-T-L

www.skylit.com
sales@skylit.com
Toll free: 888-476-1940
Fax: 978-475-1431