



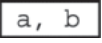
AP Computer Science Principles Exam Reference Sheet

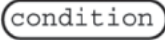
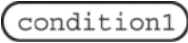
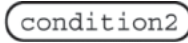

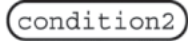

As AP Computer Science Principles does not designate any particular programming language, this reference sheet provides instructions and explanations to help students understand the format and meaning of the questions they will see on the exam. The reference sheet includes two programming formats: text based and block based.

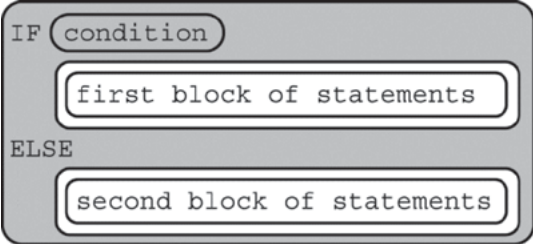
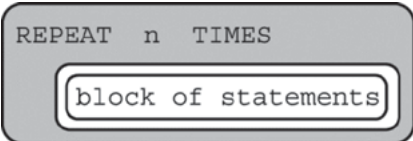

Programming instructions use four data types: numbers, Booleans, strings, and lists.

Instructions from any of the following categories may appear on the exam:





- ▶ Assignment, Display, and Input
- ▶ Arithmetic Operators and Numeric Procedures
- ▶ Relational and Boolean Operators
- ▶ Selection
- ▶ Iteration
- ▶ List Operations
- ▶ Procedures
- ▶ Robot

Instruction	Explanation
Assignment, Display, and Input	
<p>Text: a ← expression</p> <p>Block: </p>	<p>Evaluates expression and assigns the result to the variable a.</p>
<p>Text: DISPLAY (expression)</p> <p>Block: </p>	<p>Displays the value of expression, followed by a space.</p>
<p>Text: INPUT ()</p> <p>Block: INPUT</p>	<p>Accepts a value from the user and returns it.</p>
Arithmetic Operators and Numeric Procedures	
<p>Text and Block: a + b a - b a * b a / b</p>	<p>The arithmetic operators +, -, *, and / are used to perform arithmetic on a and b.</p> <p>For example, 3 / 2 evaluates to 1.5.</p>
<p>Text and Block: a MOD b</p>	<p>Evaluates to the remainder when a is divided by b. Assume that a and b are positive integers.</p> <p>For example, 17 MOD 5 evaluates to 2.</p>
<p>Text: RANDOM (a, b)</p> <p>Block: RANDOM </p>	<p>Evaluates to a random integer from a to b, including a and b.</p> <p>For example, RANDOM (1, 3) could evaluate to 1, 2, or 3.</p>
Relational and Boolean Operators	
<p>Text and Block: a = b a ≠ b a > b a < b a ≥ b a ≤ b</p>	<p>The relational operators =, ≠, >, <, ≥, and ≤ are used to test the relationship between two variables, expressions, or values.</p> <p>For example, a = b evaluates to true if a and b are equal; otherwise, it evaluates to false.</p>

Instruction	Explanation
Relational and Boolean Operators (continued)	
Text: NOT condition Block: NOT  condition	Evaluates to true if condition is false; otherwise evaluates to false.
Text: condition1 AND condition2 Block:  condition1 AND  condition2	Evaluates to true if both condition1 and condition2 are true; otherwise, evaluates to false.
Text: condition1 OR condition2 Block:  condition1 OR  condition2	Evaluates to true if condition1 is true or if condition2 is true or if both condition1 and condition2 are true; otherwise, evaluates to false.
Selection	
Text: IF (condition) { <block of statements> } Block: 	The code in block of statements is executed if the Boolean expression condition evaluates to true; no action is taken if condition evaluates to false.

Instruction	Explanation
Selection (continued)	
<p>Text:</p> <pre>IF (condition) { <first block of statements> } ELSE { <second block of statements> }</pre> <p>Block:</p> 	<p>The code in first block of statements is executed if the Boolean expression condition evaluates to true; otherwise, the code in second block of statements is executed.</p>
Iteration	
<p>Text:</p> <pre>REPEAT n TIMES { <block of statements> }</pre> <p>Block:</p> 	<p>The code in block of statements is executed n times.</p>
<p>Text:</p> <pre>REPEAT UNTIL (condition) { <block of statements> }</pre> <p>Block:</p> 	<p>The code in block of statements is repeated until the Boolean expression condition evaluates to true.</p>

Instruction	Explanation
List Operations	
For all list operations, if a list index is less than 1 or greater than the length of the list, an error message is produced and the program terminates.	
Text: <code>list[i]</code> Block: <code>list [i]</code>	Refers to the element of <code>list</code> at index <code>i</code> . The first element of <code>list</code> is at index 1.
Text: <code>list[i] ← list[j]</code> Block: <code>list [i] ← list [j]</code>	Assigns the value of <code>list[j]</code> to <code>list[i]</code> .
Text: <code>list ← [value1, value2, value3]</code> Block: <code>list ← [value1, value2, value3]</code>	Assigns <code>value1</code> , <code>value2</code> , and <code>value3</code> to <code>list[1]</code> , <code>list[2]</code> , and <code>list[3]</code> , respectively.
Text: FOR EACH item IN list { <block of statements> } Block: <code>FOR EACH item IN list</code> <code> block of statements</code>	The variable <code>item</code> is assigned the value of each element of <code>list</code> sequentially, in order from the first element to the last element. The code in block of statements is executed once for each assignment of <code>item</code> .
Text: <code>INSERT (list, i, value)</code> Block: <code>INSERT list, i, value</code>	Any values in <code>list</code> at indices greater than or equal to <code>i</code> are shifted to the right. The length of <code>list</code> is increased by 1, and <code>value</code> is placed at index <code>i</code> in <code>list</code> .
Text: <code>APPEND (list, value)</code> Block: <code>APPEND list, value</code>	The length of <code>list</code> is increased by 1, and <code>value</code> is placed at the end of <code>list</code> .

Instruction	Explanation
Robot	
If the robot attempts to move to a square that is not open or is beyond the edge of the grid, the robot will stay in its current location and the program will terminate.	
Text: <code>MOVE_FORWARD ()</code> Block: 	The robot moves one square forward in the direction it is facing.
Text: <code>ROTATE_LEFT ()</code> Block: 	The robot rotates in place 90 degrees counterclockwise (i.e., makes an in-place left turn).
Text: <code>ROTATE_RIGHT ()</code> Block: 	The robot rotates in place 90 degrees clockwise (i.e., makes an in-place right turn).
Text: <code>CAN_MOVE (direction)</code> Block: <code>CAN_MOVE</code> 	Evaluates to <code>true</code> if there is an open square one square in the direction relative to where the robot is facing; otherwise evaluates to <code>false</code> . The value of <code>direction</code> can be <code>left</code> , <code>right</code> , <code>forward</code> , or <code>backward</code> .

Appendix: Changes to the Course and Exam Description

Since the publication's update in fall 2016 some modifications have been made to the AP Computer Science Principles curriculum framework. The chart below summarizes the changes made, which are now reflected in this course and exam description.

Revision	Explanation
1. Definition of computing innovation	A computing innovation is an innovation that includes a computer or program code as an integral part of its functionality.
2. Guidance to build collaboration throughout the course	This information has been revised and moved from the assessment section into the instructional approaches section to provide guidance on building collaboration throughout the course.
3. Clarification on in-text citation	Students must support their claims with at least three in-text citations within the written responses submitted for the Explore performance task.
4. Guidance on the development and selection of algorithms and abstractions.	The algorithms (for prompt 2c) and the abstraction (for prompt 2d) selected as part of the written responses for the Create performance task must be developed independently, not collaboratively with a peer(s).
5. Organization of performance tasks guidelines	The following changes were made to the performance tasks guidelines: <ul style="list-style-type: none"> ▶ Teacher guidelines (i.e., what teachers “should” do to prepare their students, and what teachers “must,” “may,” and “may not” do during the administration of the performance tasks) have been moved to appear after the task guidelines for each performance task. ▶ Additional guidelines have been added for clarification.
6. Reproducibles for Students section	The following changes were made to the Reproducibles for Students section: <ul style="list-style-type: none"> ▶ Guidelines for preparing for the performance tasks (i.e., what students “must,” “may,” and “may not” do during the administration of each performance task) have been moved to appear after the task directions for each performance task. ▶ Additional guidelines have been added for clarification and to mirror the guidelines provided to teachers.
7. Sample Questions	All sample questions have been aligned to only one learning objective.

Contact Us

AP Services

P.O. Box 6671
Princeton, NJ 08541-6671
609-771-7300
888-225-5427 (toll free in the
U.S. and Canada)
610-290-8979 (fax)
Email: apexams@info.collegeboard.org

National Office

250 Vesey Street
New York, NY 10281
212-713-8000

AP Canada Office

2950 Douglas Street, Suite 550
Victoria, BC, Canada V8T 4N4
250-472-8561
800-667-4548 (toll free in Canada only)
Email: gewonus@ap.ca

International Services

Serving all countries outside the U.S. and Canada
250 Vesey Street
New York, NY 10281
212-373-8738
Email: international@collegeboard.org

Middle States Regional Office

*Serving Delaware, District of Columbia, Maryland,
New Jersey, New York, Pennsylvania, Puerto Rico,
and the U.S. Virgin Islands*
Three Bala Plaza East
Suite 501
Bala Cynwyd, PA 19004-1501
610-227-2550
866-392-3019
610-227-2580 (fax)
Email: msro@info.collegeboard.org

Midwestern Regional Office

*Serving Illinois, Indiana, Iowa, Kansas, Michigan,
Minnesota, Missouri, Nebraska, North Dakota,
Ohio, South Dakota, West Virginia, and Wisconsin*
6111 N. River Road, Suite 550
Rosemont, IL 60018-5158
866-392-4086
847-653-4528 (fax)
Email: mro@info.collegeboard.org

New England Regional Office

*Serving Connecticut, Maine, Massachusetts, New
Hampshire, Rhode Island, and Vermont*
1601 Trapelo Road, Suite 12
Waltham, MA 02451-1982
866-392-4089
781-663-2743 (fax)
Email: nero@info.collegeboard.org

Southern Regional Office

*Serving Alabama, Florida, Georgia, Kentucky,
Louisiana, Mississippi, North Carolina, South
Carolina, Tennessee, and Virginia*
3700 Crestwood Parkway NW, Suite 700
Duluth, GA 30096-7155
866-392-4088
770-225-4062 (fax)
Email: sro@info.collegeboard.org

Southwestern Regional Office

Serving Arkansas, New Mexico, Oklahoma, and Texas
4330 Gaines Ranch Loop, Suite 200
Austin, TX 78735-6735
866-392-3017
512-721-1841 (fax)
Email: swro@info.collegeboard.org

Western Regional Office

*Serving Alaska, Arizona, California, Colorado,
Hawaii, Idaho, Montana, Nevada, Oregon, Utah,
Washington, and Wyoming*
2099 Gateway Place, Suite 550
San Jose, CA 95110-1051
866-392-4078
408-367-1459 (fax)
Email: wro@info.collegeboard.org

