



AP[®] Computer Science A 2013 Scoring Guidelines

The College Board

The College Board is a mission-driven not-for-profit organization that connects students to college success and opportunity. Founded in 1900, the College Board was created to expand access to higher education. Today, the membership association is made up of over 6,000 of the world's leading educational institutions and is dedicated to promoting excellence and equity in education. Each year, the College Board helps more than seven million students prepare for a successful transition to college through programs and services in college readiness and college success — including the SAT[®] and the Advanced Placement Program[®]. The organization also serves the education community through research and advocacy on behalf of students, educators, and schools. The College Board is committed to the principles of excellence and equity, and that commitment is embodied in all of its programs, services, activities, and concerns.

© 2013 The College Board. College Board, Advanced Placement Program, AP, SAT and the acorn logo are registered trademarks of the College Board. All other products and services may be trademarks of their respective owners.

Visit the College Board on the Web: www.collegeboard.org.

AP Central is the official online home for the AP Program: apcentral.collegeboard.org.



AP[®] COMPUTER SCIENCE A

2013 SCORING GUIDELINES

Question 1: SongList

Part (a)	<code>getDownloadInfo</code>	4 points
-----------------	------------------------------	-----------------

Intent: Search download list for requested title and return matching `DownloadInfo` object if found.

- +1 Accesses all necessary entries in `downloadList` (no bounds errors)
- +3 Identifies and returns matching entry in `downloadList`, if it exists
 - +1 Calls `getTitle` on `DownloadInfo` object from `downloadList`
 - +1 Checks for equality between title from list object and `title` parameter (must use *String* equality check)
 - +1 Returns reference to matching object if present; `null` if not (point not awarded for early return)

Part (b)	<code>updateDownloads</code>	5 points
-----------------	------------------------------	-----------------

Intent: Update `downloadList` with information from list of titles

- +1 Accesses all entries in `titles` (no bounds error for `titles`)
- +1 Calls `getDownloadInfo(title)` to determine whether title from `titles` list exists in `downloadList`
- +1 Increments the count in matching `DownloadInfo` object if title is in `downloadList`
- +1 Constructs new `DownloadInfo` object (with correct information) if title is not in `downloadList` (point not awarded if incremented at time of construction)
- +1 Adds constructed object to end of `downloadList` if title is not in `downloadList` (point not awarded if added more than once)

Question-Specific Penalties

- 1 (g) Uses `getLength/getSize` for `ArrayList` size
- 2 (v) Consistently uses incorrect array name instead of `downloadList/titles`
- 1 (z) Attempts to return a value from `updateDownloads`

AP[®] COMPUTER SCIENCE A

2013 SCORING GUIDELINES

Question 2: TokenPass

Part (a)	TokenPass constructor	4 points
-----------------	-----------------------	-----------------

Intent: Create TokenPass object and correctly initialize game state

- +1 Creates instance variable `board` as `int` array of size `playerCount`
- +1 Computes a random number between 1 and 10, inclusive, and a random number between 0 and `playerCount-1`, inclusive
- +1 Initializes all entries in `board` with computed random value (*no bounds errors*)
- +1 Initializes instance variable `currentPlayer` to computed random value

Part (b)	<code>distributeCurrentPlayerTokens</code>	5 points
-----------------	--	-----------------

Intent: Distribute all tokens from `currentPlayer` position to subsequent positions in array

- +1 Uses initial value of `board[currentPlayer]` to control distribution of tokens
- +1 Increases at least one `board` entry in the context of a loop
- +1 Starts distribution of tokens at correct board entry
- +1 Distributes next token (if any remain) to position 0 after distributing to highest position in board
- +1 On exit: token count at each position in `board` is correct

Question-Specific Penalties

- 2 (v) Consistently uses incorrect array name instead of `board`
- 1 (y) Destruction of persistent data (`currentPlayer`)
- 1 (z) Attempts to return a value from `distributeCurrentPlayerTokens`

AP[®] COMPUTER SCIENCE A 2013 SCORING GUIDELINES

Question 3: JumpingCritic (GridWorld)

Part (a)	<code>getEmptyLocations</code>	5 points
-----------------	--------------------------------	-----------------

Intent: *Create and return* `ArrayList<Location>` *of all empty locations in* `grid`

- +½ Declares and constructs empty `ArrayList<Location>`
- +½ Accesses all locations in `grid` (*no bounds errors*)
- +2 Identifies empty location in grid in context of loop
 - +1 Creates new location in grid
 - +1 Determines if created location is empty
- +1 Includes all and only identified empty locations in constructed arraylist exactly once
- +1 Returns the constructed arraylist (*code must have examined grid*)

Part (b)	Class: <code>JumpingCritic</code>	4 points
-----------------	-----------------------------------	-----------------

Intent: *Define extension to* `Critic` *class that jumps to randomly selected empty location in* `its grid`

- +½ `class JumpingCritic extends Critter`
- +1½ Override `getMoveLocations`
 - +½ `public ArrayList<Location> getMoveLocations()`
 - +½ `GridWorldUtilities.getEmptyLocations(getGrid())`
 - +½ Returns arraylist containing empty locations
- +1 Handles `null` location case correctly in `selectMoveLocation`
- +1 Handles random location case correctly (must override `getMoveLocations`)

Question-Specific Penalties

- 1 (s) Causes inappropriate state change in world (`Grid`, `Actor`, ...)
- 1 (t) Overrides `act`

AP[®] COMPUTER SCIENCE A

2013 SCORING GUIDELINES

Question 4: SkyView

Part (a)	SkyView constructor	5 points
-----------------	---------------------	-----------------

Intent: Construct `SkyView` object from 1D array of scan data

- +1 Constructs correctly-sized 2D array of doubles and assigns to instance variable `view`
- +1 Initializes at least one element of `view` with value from element of `scanned` (*must be in context of loop*)
- +1 Places consecutive values from `scanned` into at least one row of `view` in original order
- +1 Places consecutive values from `scanned` into at least one row of `view` in reverse order
- +1 On exit: all elements of `view` have correct values (*no bounds errors on `view` or `scanned`*)

Part (b)	<code>getAverage</code>	4 points
-----------------	-------------------------	-----------------

Intent: Compute and return average of rectangular section of `view`, specified by parameters

- +1 Declares and initializes a `double` accumulator
- +1 Adds all and only necessary values from `view` to accumulator (*no bounds errors*)
- +1 Computes average of specified rectangular section
- +1 Returns the computed average (*computation must involve `view`*)

Question-Specific Penalties

- 2 (v) Consistently uses incorrect array name instead of `view/scanned`