

AP® Computer Science A
2006 Free-Response Questions

The College Board: Connecting Students to College Success

The College Board is a not-for-profit membership association whose mission is to connect students to college success and opportunity. Founded in 1900, the association is composed of more than 5,000 schools, colleges, universities, and other educational organizations. Each year, the College Board serves seven million students and their parents, 23,000 high schools, and 3,500 colleges through major programs and services in college admissions, guidance, assessment, financial aid, enrollment, and teaching and learning. Among its best-known programs are the SAT®, the PSAT/NMSQT®, and the Advanced Placement Program® (AP®). 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.

© 2006 The College Board. All rights reserved. College Board, AP Central, APCD, Advanced Placement Program, AP, AP Vertical Teams, Pre-AP, SAT, and the acorn logo are registered trademarks of the College Board. Admitted Class Evaluation Service, CollegeEd, connect to college success, MyRoad, SAT Professional Development, SAT Readiness Program, and Setting the Cornerstones are trademarks owned by the College Board. PSAT/NMSQT is a registered trademark of the College Board and National Merit Scholarship Corporation. All other products and services may be trademarks of their respective owners. Permission to use copyrighted College Board materials may be requested online at: www.collegeboard.com/inquiry/cbpermit.html.

Visit the College Board on the Web: www.collegeboard.com.
AP Central is the official online home for the AP Program: apcentral.collegeboard.com.

COMPUTER SCIENCE A
SECTION II
Time—1 hour and 45 minutes
Number of questions—4
Percent of total grade—50

Directions: SHOW ALL YOUR WORK. REMEMBER THAT PROGRAM SEGMENTS ARE TO BE WRITTEN IN Java.

Notes:
- Assume that the classes listed in the Quick Reference found in the Appendix have been imported where appropriate.
- Unless otherwise noted in the question, assume that parameters in method calls are not `null` and that methods are called only when their preconditions are satisfied.
- In writing solutions for each question, you may use any of the accessible methods that are listed in classes defined in that question. Writing significant amounts of code that can be replaced by a call to one of these methods may not receive full credit.

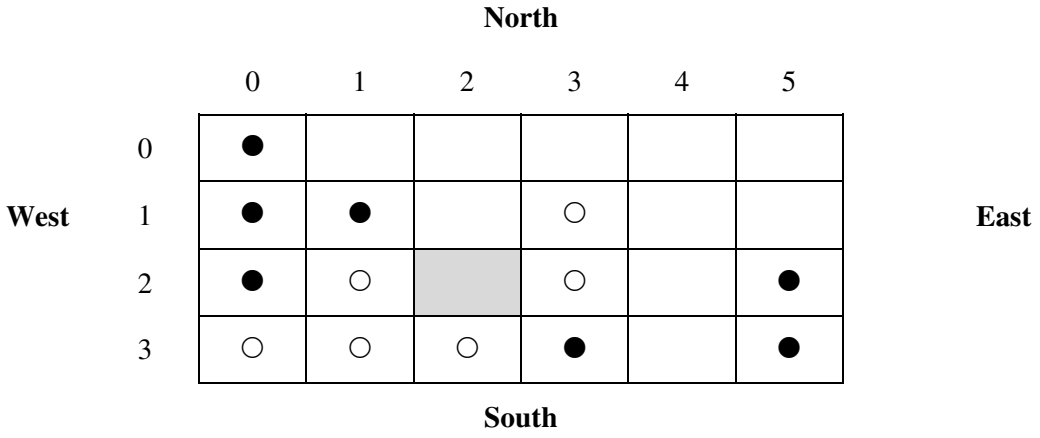
© 2006 The College Board. All rights reserved.
Visit apcentral.collegeboard.com (for AP professionals) and www.collegeboard.com/apstudents (for students and parents).

4. This question involves reasoning about the code from the Marine Biology Simulation case study. A copy of the code is provided as part of this exam.

   Consider using the `BoundedEnv` class from the Marine Biology Simulation case study to model a game board. In this implementation of the `Environment` interface, each location has at most **four** neighbors. Those neighbors are determined by the `Environment` method `neighborsOf`.

   DropGame is a two-player game that is played on a rectangular board. The players — designated as BLACK and WHITE — alternate, taking turns dropping a colored piece in a column. A dropped piece will fall down the chosen column until it comes to rest in the empty location with the largest row index. If the location for the **newly dropped** piece has **three** neighbors that match its color, the player that dropped this piece wins the game.

   The diagram below shows a sample game board on which several moves have been made.



   The following chart shows where a piece dropped in each column would land on this board.

Column	Location for Piece Dropped in the Column
0	No piece can be placed, since the column is full
1	(0, 1)
2	(2, 2)
3	(0, 3)
4	(3, 4)
5	(1, 5)

   Note that a WHITE piece dropped in column 2 would land in the shaded cell at location (2, 2) and result in a win for WHITE because the three neighboring locations — (2, 1), (3, 2), and (2, 3) — contain WHITE pieces. This move is the only available winning move on the above game board. Note that a BLACK piece dropped in column 1 would land in location (0, 1) and not result in a win because the neighboring location (0, 2) does not contain a BLACK piece.

© 2006 The College Board. All rights reserved.
Visit apcentral.collegeboard.com (for AP professionals) and www.collegeboard.com/apstudents (for students and parents).

The `Piece` class implements the `Locatable` interface and is defined as follows.

```
public class Piece implements Locatable
{
  // returns location of this Piece
  public Location location()
  {  /* implementation not shown */  }

  // returns color of this Piece
  public Color color()
  {  /* implementation not shown */  }

  // There may be fields, constructors, and methods that are not shown.
}
```

An incomplete definition of the `DropGame` class is shown below. The class contains a private instance variable `theEnv` to refer to the `Environment` that represents the game board. Players will add `Piece` objects to this environment as they take turns. You will implement two methods for the `DropGame` class.

```
public class DropGame
{
  private Environment theEnv;  // contains Piece objects


  // returns null if no empty locations in column;
  // otherwise, returns the empty location with the
  // largest row index within the specified column;
  // precondition: 0 <= column < theEnv.numCols()
  public Location dropLocationForColumn(int column)
  {  /* to be implemented in part (a) */  }


  // returns true if dropping a piece of the given color into the
  // specified column matches color with three neighbors;
  // otherwise, returns false
  // precondition: 0 <= column < theEnv.numCols()
  public boolean dropMatchesNeighbors(int column, Color pieceColor)
  {  /* to be implemented in part (b) */  }


  // There may be fields, constructors, and methods that are not shown.
}
```

© 2006 The College Board. All rights reserved.
Visit apcentral.collegeboard.com (for AP professionals) and www.collegeboard.com/apstudents (for students and parents).

(a) Write the `DropGame` method `dropLocationForColumn`, which returns the resulting `Location` for a piece dropped into the specified column. If there are no empty locations in the column, the method should return `null`. Otherwise, of the empty locations in the column, the location with the largest row index should be returned.

In writing `dropLocationForColumn`, you may use any methods defined in the `DropGame` class or accessible methods of the case study classes.

Complete method `dropLocationForColumn` below.

```
// returns null if no empty locations in column;
// otherwise, returns the empty location with the
// largest row index within the specified column;
// precondition: 0 <= column < theEnv.numCols()
public Location dropLocationForColumn(int column)
```

(b) Write the `DropGame` method `dropMatchesNeighbors`, which returns `true` if dropping a piece of a given color into a specific column will match the color of three of its neighbors. The location to be checked for matches with its neighbors is the location identified by method `dropLocationForColumn`. If there are no empty locations in the column, `dropMatchesNeighbors` returns `false`.

In writing `dropMatchesNeighbors`, you may assume that `dropLocationForColumn` works as specified regardless of what you wrote in part (a).

Complete method `dropMatchesNeighbors` below.

```
// returns true if dropping a piece of the given color into the
// specified column matches color with three neighbors;
// otherwise, returns false
// precondition: 0 <= column < theEnv.numCols()
public boolean dropMatchesNeighbors(int column, Color pieceColor)
```

END OF EXAM

© 2006 The College Board. All rights reserved.
Visit apcentral.collegeboard.com (for AP professionals) and www.collegeboard.com/apstudents (for students and parents).