

AP® Computer Science A
2008 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.

© 2008 The College Board. All rights reserved. College Board, Advanced Placement Program, AP, AP Central, SAT, and the acorn logo are registered trademarks of the College Board. PSAT/NMSQT is a registered trademark of the College Board and National Merit Scholarship Corporation.
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.

1. A travel agency maintains a list of information about airline flights. Flight information includes a departure time and an arrival time. You may assume that the two times occur on the same day. These times are represented by objects of the `Time` class.

   The declaration for the `Time` class is shown below. It includes a method `minutesUntil` that returns the difference (in minutes) between the current `Time` object and another `Time` object.

```
public class Time
{
  /** @return difference, in minutes, between this time and other;
   *          difference is negative if other is earlier than this time
   */
  public int minutesUntil(Time other)
  {  /* implementation not shown */  }

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

   For example, assume that `t1` and `t2` are `Time` objects where `t1` represents 1:00 P.M. and `t2` represents 2:15 P.M. The call `t1.minutesUntil(t2)` will return `75` and the call `t2.minutesUntil(t1)` will return `-75`.

© 2008 The College Board. All rights reserved.
Visit apcentral.collegeboard.com (for AP professionals) and www.collegeboard.com/apstudents (for students and parents).

4. A *checker* is an object that examines strings and *accepts* those strings that meet a particular criterion.

The `Checker` interface is defined below.

```
public interface Checker
{
  /** @param text a string to consider for acceptance
   *  @return true if this Checker accepts text; false otherwise
   */
  boolean accept(String text);
}
```

In this question, you will write two classes that implement the `Checker` interface. You will then create a `Checker` object that checks for a particular acceptance criterion.

(a) A `SubstringChecker` accepts any string that contains a particular substring. For example, the following `SubstringChecker` object `broccoliChecker` accepts all strings containing the substring `"broccoli"`.

```
Checker broccoliChecker = new SubstringChecker("broccoli");
```

The following table illustrates the results of several calls to the `broccoliChecker accept` method.

Method Call	Result
`broccoliChecker.accept("broccoli")`	`true`
`broccoliChecker.accept("I like broccoli")`	`true`
`broccoliChecker.accept("carrots are great")`	`false`
`broccoliChecker.accept("Broccoli Bonanza")`	`false`

Write the `SubstringChecker` class that implements the `Checker` interface. The constructor should take a single `String` parameter that represents the particular substring to be matched.

© 2008 The College Board. All rights reserved.
Visit apcentral.collegeboard.com (for AP professionals) and www.collegeboard.com/apstudents (for students and parents).

(b) Checkers can be created to check for multiple acceptance criteria by combining other checker objects. For example, an `AndChecker` is a `Checker` that is constructed with two objects of classes that implement the `Checker` interface (such as `SubstringChecker` or `AndChecker` objects). The `AndChecker accept` method returns `true` if and only if the string is accepted by both of the `Checker` objects with which it was constructed.

In the code segment below, the `bothChecker` object accepts all strings containing both `"beets"` and `"carrots"`. The code segment also shows how the `veggies` object can be constructed to accept all strings containing the three substrings `"beets"`, `"carrots"`, and `"artichokes"`.

```
Checker bChecker = new SubstringChecker("beets");
Checker cChecker = new SubstringChecker("carrots");
Checker bothChecker = new AndChecker(bChecker, cChecker);

Checker aChecker = new SubstringChecker("artichokes");
Checker veggies = new AndChecker(bothChecker, aChecker);
```

The following table illustrates the results of several calls to the `bothChecker accept` method and the `veggies accept` method.

Method Call	Result
`bothChecker.accept("I love beets and carrots")`	`true`
`bothChecker.accept("beets are great")`	`false`
`veggies.accept("artichokes, beets, and carrots")`	`true`

Write the `AndChecker` class that implements the `Checker` interface. The constructor should take two `Checker` parameters.

© 2008 The College Board. All rights reserved.
Visit apcentral.collegeboard.com (for AP professionals) and www.collegeboard.com/apstudents (for students and parents).

(c) Another implementation of the `Checker` interface is the `NotChecker`, which contains the following:

- A one-parameter constructor that takes one `Checker` object
- An `accept` method that returns `true` if and only if its `Checker` object does NOT accept the string

Using any of the classes `SubstringChecker`, `AndChecker`, and `NotChecker`, construct a `Checker` that accepts a string if and only if it contains neither the substring `"artichokes"` nor the substring `"kale"`. Assign the constructed `Checker` to `yummyChecker`. Consider the following incomplete code segment.

```
Checker aChecker = new SubstringChecker("artichokes");
Checker kChecker = new SubstringChecker("kale");
Checker yummyChecker;
/* code to construct and assign to yummyChecker */
```

The following table illustrates the results of several calls to the `yummyChecker accept` method.

Method Call	Result
`yummyChecker.accept("chocolate truffles")`	`true`
`yummyChecker.accept("kale is great")`	`false`
`yummyChecker.accept("Yuck: artichokes & kale")`	`false`

In writing your solution, you may use any of the classes specified for this problem. Assume that these classes work as specified, regardless of what you wrote in parts (a) and (b). You may assume that the declarations for `aChecker`, `kChecker`, and `yummyChecker` in the code segment above have already been executed.

Write your `/* code to construct and assign to yummyChecker */` below.

STOP

END OF EXAM

© 2008 The College Board. All rights reserved.
Visit apcentral.collegeboard.com (for AP professionals) and www.collegeboard.com/apstudents (for students and parents).