Results from the 2017 AP Computer Science A Exam

John Cigas, ap@cigas.net

AP CS A Chief Reader

Park University

July 29, 2017

CollegeBoard AP

AP® Annual Conference 2017

Outline

- **The Course**
- **The Reading**
- **The Exam**
- **The Results**
- **Recommendations**
- **Resources**
- **Questions**

Description

The AP Computer Science A course introduces students to computer science with fundamental topics that include:

- problem solving
- design strategies and methodologies
- organization of data (data structures)
- approaches to processing data (algorithms)
- analysis of potential solutions
- ethical and social implications of computing.

From the Computer Science A Course Description

Development

- **Development**
  - College Board
  - Curriculum and Exam
  - Development Committee
    - High school & college faculty

- **Administration and Scoring**
  - Educational Testing Service
  - Logistics
  - Readers

Exam History

- 1984 First Examination (Pascal)
- 1988 Exam split into A and AB (A subscore)
- 1992 A subscore eliminated
- 1995 Case study introduced
- 1999 C++
- 2004 Java
- 2008 GridWorld case study introduced
- 2009 Last year of AB exam
- 2014 Last year of GridWorld Case Study
- 2015 AP CS A Labs introduced
- 2016 Time rebalanced in exam
- 2017 CS Principles launches

AP Computer Science A 2017

The Reading

CollegeBoard AP

AP® Annual Conference 2017

Process

Off site

- **Last year**
  - Free-response questions
  - Preliminary rubrics
- **This May**
  - Chief Reader
    - Rubrics
    - Scoring notes
    - Canonical solutions
  - Exam Leaders & College Board CIA
    - Refine these components

Process

Onsite - Prereading

- **Pre-reading**
  - Question Leaders
    - Vet the rubrics
    - Develop reader training
    - Train the table leaders
  - Table Leaders
    - Vet the training
    - Prepare mentoring support for readers

Process

Onsite - Reading

- **Reading**
  - Question Leaders
    - Train readers
  - Table Leaders
    - Mentor readers
    - Assist in applying the rubric
  - Readers read!
    - 8:00 am – 5:00 pm for 7 days
    - 7.5 hours reading exams, 2x15-minute break, 1-hour lunch

Process

Reader Training

- **Consistency checks**
  - Training packs
  - Partner system
  - Split packs
  - Check-reading
  - Reading Management System (RMS) metrics
- **Operational exam readers usually read one question the entire reading**

Process

Other Activities

- **Readers do more than read**
  - Social lounge (hotel)
  - Opening night reception
  - Toy night (Pedagogical Practices)
  - Meet the DC and College Board Forum
  - Professional Night
  - Dine out night

Stats

	2017	2012
Exams	~61,000	~22,500
Readers	256	119
Table Leaders	33	17
Question Leaders	19	17
Exam Leaders	2	2

AP Computer Science A 2017

The Exam

CollegeBoard AP

AP® Annual Conference 2017

The Exam

Appendix A: AP Java Subset

- **Outlines the features of Java that may appear on the AP Computer Science A Exam.**
- **No free response questions require constructs outside the subset.**
- **The AP Java subset is NOT intended to restrict content of courses**
- **The subset itself will need to be supplemented in order to address all topics in a typical introductory curriculum.**

The Exam

Solutions

- **Solutions not restricted to the subset**
- **All correct solutions earn full credit***
- **Solutions may utilize any standard Java constructs or classes ***
- **Some minor errors are ignored**

***Unless otherwise prohibited**

The 2017 Exam

- **Multiple choice was 1.5 hours**
- **Free response was 1.5 hours**
  - Students had 22.5 minutes per free response question to
    - Read
    - Understand
    - Design/solve
    - Code
    - Check solutions

2017 Exam Questions

- **Q1: Digits**
  - Extract individual digits from an integer
  - Store and process these in an ArrayList
- **Q2: Multiplication Practice**
  - Design a class for generating arithmetic practice drills
- **Q3: Phrase Editor**
  - Use String methods from the AP Java subset to find and replace contents of a string
- **Q4: Successor Array**
  - Search a 2D array of integers
  - Create a 2D array of Position objects

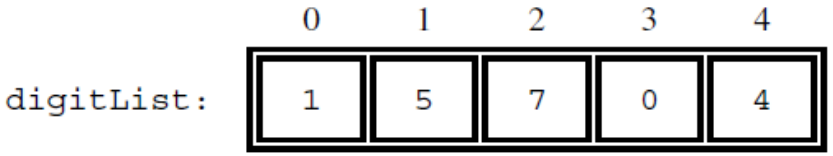
Q1: Digits part (a)

(a) Write the constructor for the `Digits` class. The constructor initializes and fills `digitList` with the digits from the non-negative integer `num`. The elements in `digitList` must be `Integer` objects representing single digits, and appear in the same order as the digits in `num`. Each of the following examples shows the declaration of a `Digits` object and the contents of `digitList` as initialized by the constructor.

Example 1

```
Digits d1 = new Digits(15704);
```

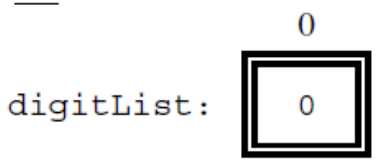
`d1`:



Example 2

```
Digits d2 = new Digits(0);
```

`d2`:



Q1: Digits part (a)

Example Solution

```
public Digits(int num)
{
    digitList = new ArrayList<Integer>();

    if (num == 0)
        digitList.add(new Integer(0));

    while(num > 0)
    {
        digitList.add(0, new Integer(num % 10));
        num /= 10;
    }
}
```

Q1: Digits part (a)

Scoring Rubric

Part (a) Digits constructor **5 points**

Intent: *Initialize instance variable using passed parameter*

+1 Constructs `digitList`

+1 Identifies a digit in `num`

+1 Adds at least one identified digit to a list

+1 Adds all identified digits to a list (*must be in context of a loop)*

+1 **On exit**: `digitList` contains all and only digits of num in the correct order.

Q1 – Digits part (a)

Scoring Notes

Rubric Criteria	Responses earn the point if they…	Responses will not earn the point if they…
Constructs `digitList`		• initialize a local variable instead of `digitList` • create an `ArrayList <int>`
Identifies a digit in `num`	• identify one digit of `num` or a length one substring/character of the String representation of `num`.	• treat `num` itself as a String • convert `num` to a String incorrectly
Adds at least one identified digit to a list	• call `add` for some ArrayList using the previously identified digit, even if that digit was identified incorrectly.	• add String or char to `digitList` without proper conversion to the correct type

Q1: Digits part (b)

(b) Write the `Digits` method `isStrictlyIncreasing`. The method returns `true` if the elements of `digitList` appear in strictly increasing order; otherwise, it returns `false`. A list is considered strictly increasing if each element after the first is greater than (but not equal to) the preceding element.

The following table shows the results of several calls to `isStrictlyIncreasing`.

Method call	Value returned
`new Digits(7).isStrictlyIncreasing()`	`true`
`new Digits(1356).isStrictlyIncreasing()`	`true`
`new Digits(1336).isStrictlyIncreasing()`	`false`
`new Digits(1536).isStrictlyIncreasing()`	`false`
`new Digits(65310).isStrictlyIncreasing()`	`false`

Q1 – Digits part (b)

Example Solution

```
public boolean isStrictlyIncreasing()
{
    for (int i=0; i<digitList.size()-1; i++)
    {
        if (digitList.get(i).intValue() >=
            digitList.get(i+1).intValue())
            return false;
    }
    return true;
}
```

Q1: Digits part (b)

Part (b) isStrictlyIncreasing **4 points**

Intent: *Determine whether or not elements in digitList are in increasing order*

+1 Compares at least one identified consecutive pair of `digitList` elements

+1 Determines if a consecutive pair of `digitList` is out of order (*must be in context of a* `digitList` *traversal*)

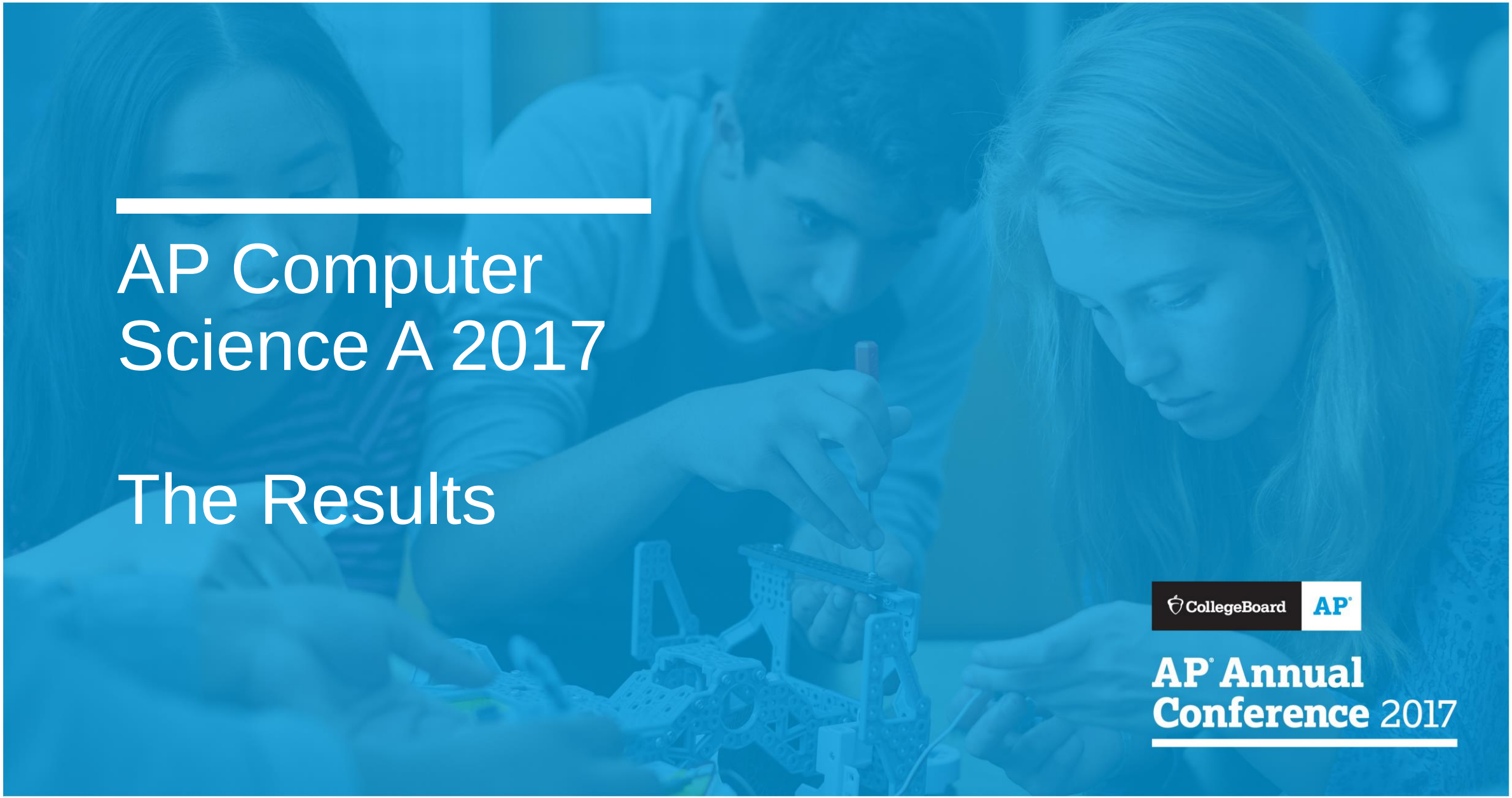
+1 Compares all necessary consecutive pairs of elements (*no bounds errors*)

+1 Returns true iff all consecutive pairs of elements are in order; returns false otherwise

Q1 – Digits part (b)

Scoring Notes

Rubric Criteria	Responses earn the point if they...	Responses will not earn the point if they...
Compares at least one identified consecutive pair of `digitList` elements	• compare two consecutive Integers using `compareTo` • explicitly convert two consecutive Integers to ints and compare those with >=, <= etc. • use auto-unboxing to convert two consecutive Integers to ints and compare those with >=, <= etc.	• access `digitList` as an array or string • fail to call .get() • compare using `!>`
Determines if a consecutive pair of `digitList` is out of order (must be in context of a `digitList` traversal)	• determine the correct relationship between the two compared consecutive elements, even if the syntax of the comparison is incorrect.	• fail to consider the case where the two elements are equal for the false case



AP Computer Science A 2017

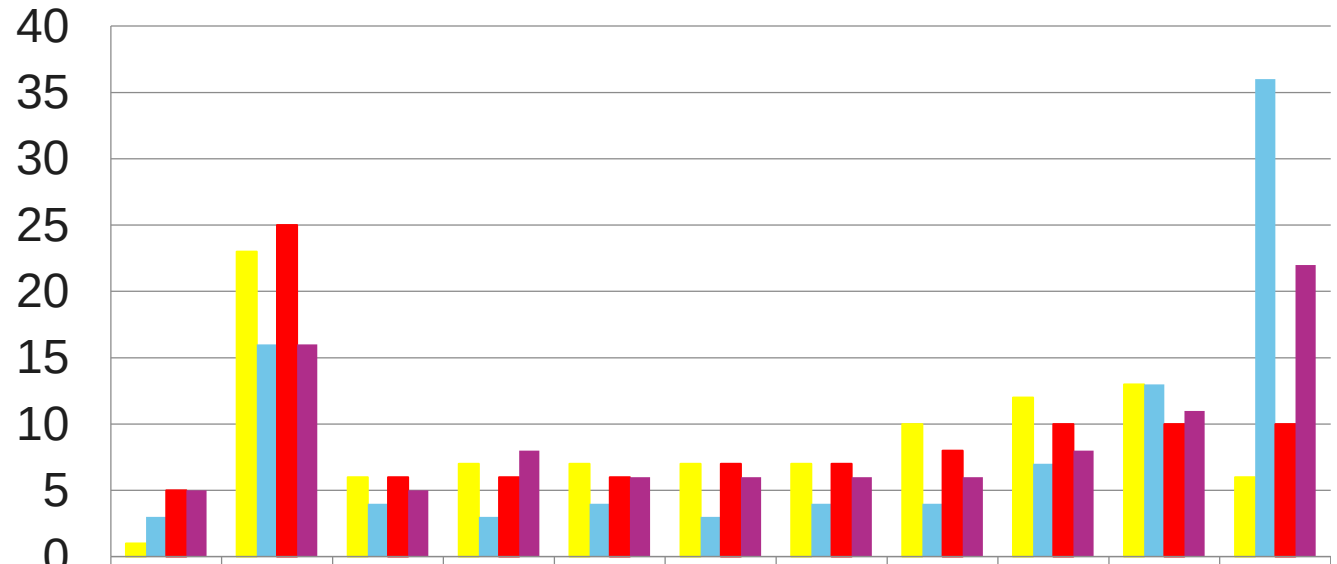
The Results

CollegeBoard AP

AP® Annual Conference 2017

Number of Exams

Percentage of Exams Receiving Raw Score



	-	0	1	2	3	4	5	6	7	8	9
Digits	1	23	6	7	7	7	7	10	12	13	6
Mult Practice	3	16	4	3	4	3	4	4	7	13	36
Phrase Editor	5	25	6	6	6	7	7	8	10	10	10
Successor Array	5	16	5	8	6	6	6	6	8	11	22

Raw Score Distribution

	Q1	Q2	Q3	Q4
Mean (no -/0)	4.10 (5.42)	5.86 (7.16)	3.84 (5.49)	4.75 (6.02)
Std Deviation	3.17	3.54	3.31	3.46

Note: The first means are official. The means with no -/0 are estimates from last day of reading.

Grade Setting

The Chief Reader, ETS content assessment specialists, College Board representatives, and ETS statisticians are involved in a grade-setting meeting where the data and analyses are presented, including comparisons with previous years, and at which the parties establish and agree to mapping raw scores to reported grades of 1 to 5.

1 = No recommendation

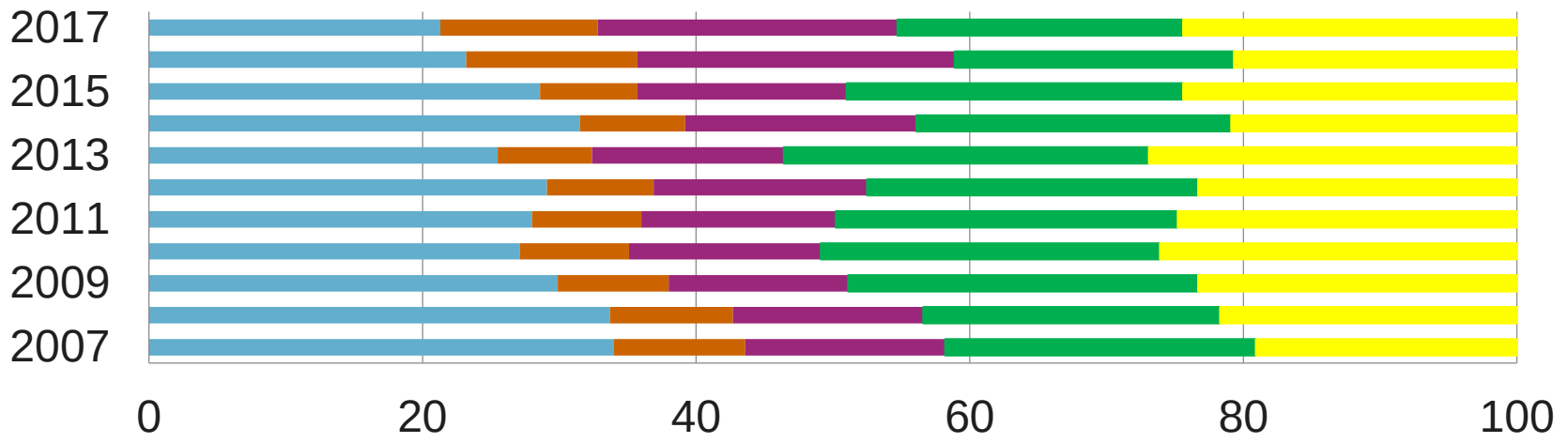
2 = Possibly qualified

3 = Qualified

4 = Well qualified

5 = Extremely well qualified

Score Distribution (1-5) (Percentage)



	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017
1	34	33.7	29.9	27.1	28	29.1	25.5	31.5	28.6	23.2	21.3
2	9.6	9	8.1	8	8	7.8	6.9	7.7	7.1	12.5	11.5
3	14.6	13.9	13.1	14	14.2	15.6	14	16.9	15.3	23.2	21.9
4	22.7	21.7	25.6	24.8	25	24.2	26.7	23	24.6	20.4	20.9
5	19.1	21.7	23.3	26.1	24.8	23.3	26.9	20.9	24.4	20.7	24.4

AP Computer Science A 2017

Recommendations

CollegeBoard AP

AP® Annual Conference 2017

Recommendations

- Discuss/practice test-taking skills

Test-Taking Skills

- Read the question
- Write legibly
- Clearly indicate answer on page
  cross out unwanted code
- Eschew obfuscation
- Use reference material
- Read part (b) even if unable to do part (a)
- Test solution using examples
- Use top-down design
  especially when time is short
- Preconditions are your friend
- Read the question

Recommendations

- Discuss/practice test-taking skills
- Address the common errors

Common Errors

- Failure to read question closely
- Reimplementation of helper functions
- Confusion of == and equals
- Use of new object without construction
- Use of enhanced for loop when not appropriate
- Confusion between lists and arrays
- Accessing too many/few elements in array/list
- Failure to test for and handle boundary cases
- Confusion as to how and when to return values (or not)
- Confusion between local and instance variables
- Failure to understand the problem abstraction

Recommendations

- Discuss/practice test-taking skills
- Address the common errors
- Use material from the reading

Available Material

- http://apcentral.collegeboard.com
- Released free response questions
- Scoring guidelines
- Student performance Q & A (question intent, common errors, recommendations to teachers)
- Sample responses and commentary
- Teacher community on AP Central
- College Board workshops/summer institutes
- Facebook group (unofficial)

Recommendations

- Discuss/practice test-taking skills
- Address the common errors
- Use material from the reading
- Become a reader

Resources

- http://apcentral.collegeboard.com
  AP Central: AP info, course descriptions, materials
- http://www.collegeboard.com
  College Board: general info about CB, the AP program
- https://apcommunity.collegeboard.org/
  Discussion groups for AP teachers
- https://www.facebook.com/groups/APComputerScienceTeachers/
  Facebook group (unofficial)
- **Email me: ap@cigas.net**

Questions?

CollegeBoard AP

AP® Annual Conference 2017