What Colleges Should Know about the New AP Computer Science

David Reed, Creighton University
Chief Reader of AP Computer Science
davereed@creighton.edu

Jody Paul, Metropolitan State College of Denver
Computer Science National Leader, College Board
jody@acm.org

Judy Hromcik, Arlington High School
Past Member, AP Computer Science Development Committee
jhromcik@aisd.net

What is AP?

the Advanced Placement (AP) Program® is a cooperative effort between secondary schools and colleges and universities

AP works with college & high school teachers to develop college-level courses for high schools (35 courses in 20 subject areas)

including: AP Computer Science A, corresponding to CS1 in colleges
AP Computer Science AB, corresponding to CS2 in colleges

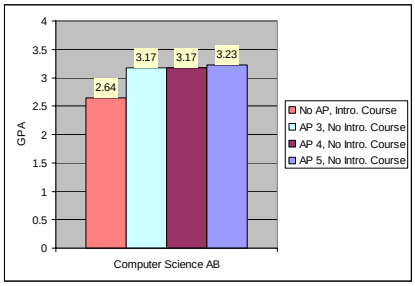
AP develops and administers standardized exams to assess student achievement, which can result in college credit

- the AP Program is offered by the College Board®
  - a nonprofit association of more than 3,900 schools, colleges, and organizations
- AP exams are developed/administered by the Educational Testing Service®
  - a nonprofit, world's largest private educational testing and measurement organization

2

Who Benefits from AP?

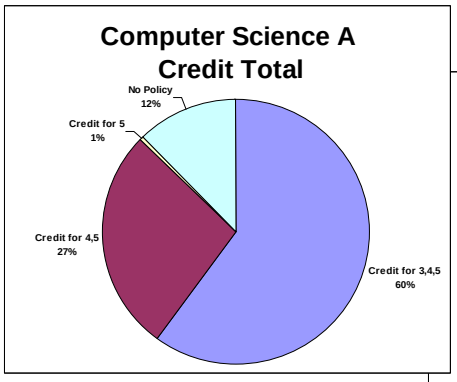
- more than 60% of U.S. high schools offer AP courses
  - more than 115,000 high school teachers
- in 2005, 1,221,016 students took 2,105,803 AP exams
- AP courses provide high school students with college-level challenges, proven curricular models, and a chance to get a head-start on college
  - research has shown that students who place out of courses in college due to AP credit subsequently perform better than peers who don't

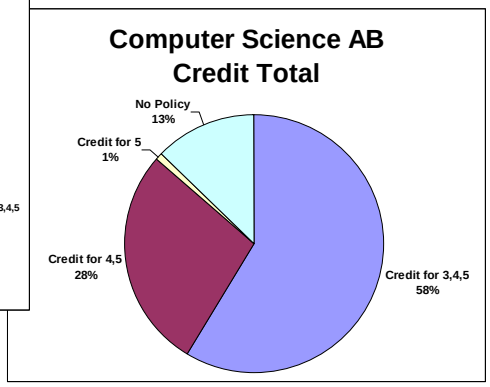


3

Who Recognizes AP?

- more than 90% of U.S. colleges and universities award credit and/or placement based on AP exam scores





APCS Exam

1984: first APCS exam, in Pascal
(6,911 exams)

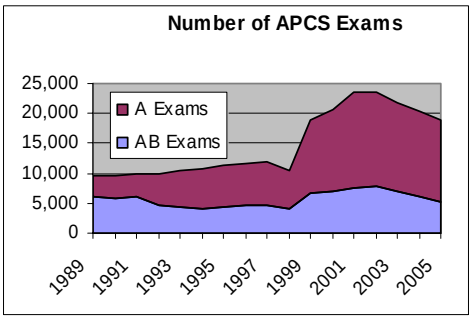
1992: split into separate A and AB exams
(5,230 + 4,643 = 9,873 exams)

1995: first case study introduced
(6,919 + 4,362 = 11,281 exams)

1999: exam language switched to C++
(12,218 + 6,619 = 18,837 exams)

2004: exam language switched to Java
(14,337 + 6.077 = 20,414 exams)

2005: 2nd year of Java
(13,924 + 5,097 = 19,021 exams)



5

College Involvement

since the program is designed to offer college-level courses for high schools, college faculty are involved in all aspects

- curriculum development, test writing and grading, grade setting, ...
- periodically, college faculty are surveyed regarding course content
  - based on a 1999-2000 survey, the APCS curriculum was revised to emphasize object-oriented methods and (a manageable subset of) Java
- *APCS Development Committee* (3 college + 3 high school faculty) is responsible for curriculum development and exam writing

  Scot Drysdale, Dartmouth College (chair)
  Cay Horstmann, San Jose State University
  Laurie White, Mercer University
  Don Allen, Troy High School (CA)
  Reg Hahne, Atholton High School (MD)
  Ann Shen, Bishop Strachan School (Toronto)
- *APCS Chief Reader* is responsible for grading the exams and equating scores with college-level performance

  David Reed, Creighton University

6

APCS Curricula

Computer Science A and AB	AB only
Program design 1. Read and understand a problem's description, purpose, and goals. 2. Apply data abstraction and encapsulation. 3. Read and understand class specifications and relationships among the classes ("is-a", "has-a" relationships). 4. Understand and implement a given class hierarchy. 5. Identify reusable components from existing code using classes and class libraries. Class design 1. Design and implement a class. 2. Design an interface. 3. Choose appropriate data representation and algorithms. 4. Apply functional decomposition. 5. Extend a given class using inheritance.	1. Specify the purpose and goals for a problem. 3. Decompose a problem into classes, define relationships and responsibilities of those classes. 1. Design and implement a set of interacting classes. 3. Choose appropriate advanced data structures and algorithms.
Implementation techniques 1. Methodology: Object-oriented development, Top-down development, Encapsulation and information hiding, Procedural abstraction Programming constructs 1. Primitive types vs. objects 2. Declaration: Constant declarations, Variable declarations, Class declarations Interface declarations, Method declarations, Parameter declarations 3. Console output (System.out.print/println) 4. Control: Methods, Sequential, Conditional, Iteration, Recursion	see apcentral.collegeboard.com for full descriptions

7

APCS Curricula (cont.)

Computer Science A and AB	AB only
Data Structures 1. Simple data types (int, boolean, double) 2. Classes 3. One-dimensional arrays 4. Java Collections: ArrayList Algorithms 1. Data Structure Operations: Traversals, Insertions, Deletions 2. Searching: Sequential, Binary 3. Sorting: Selection, Insertion, Mergesort	3. Two-dimensional arrays, Linked lists, Stacks, Queues, Trees, Priority Queues, Sets, Maps 4. Java Collections: List, ArrayList, LinkedList, Set, HashSet, TreeSet, Map, HashMap, TreeMap 1. Data Structure Operations: Iterators 2. Searching: Hashing 3. Sorting: QuickSort, Heapsort
Testing 1. Test classes and libraries in isolation 2. Identify boundary cases and generate appropriate test data 3. Perform integration testing Debugging 1. Categorize errors: compile-time, run-time, logic 2. Identify and correct errors (use debugger, output statements, hand-trace code) Analysis of algorithms 1. Informal comparisons of running times, 2. calculation of exact statement execution counts Understand error handling, run-time exceptions, Reason about Programs 1. pre- and post- conditions, assertions, numerical representations and limits	3. Big-Oh notation 4. Worst-case and average-case time and space analysis Throw runtime exceptions 1. Invariants

8

APCS Exams

in addition, APCS students are required to know a large case study

- the Marine Biology Case Study consists of > 10 interacting classes
- accompanying narrative discusses design choices & implementation details

my personal observation: the A & AB curricula are far more rigorous than most college CS1 & CS2 courses

each AP exam consists of

- 40 multiple choice questions (~4-6 of which relate to the case study)
- 4 free response questions (1 of which relates to the case study)

  free response questions may involve:
  - implementing an algorithm
  - completing methods in an existing class
  - designing and implementing classes in an inheritance hierarchy
  - selecting, implementing, and analyzing data structures
  - . . .

9

Performance Assessment

Jody will talk about the grading process

- multiple choice graded by computer; free response graded by teachers
- the Chief Reader is responsible for assigning final scores (1 – 5)

- periodically, comparability studies are performed to equate scores with college-level performance
  - representative colleges administer sections of the exam in actual courses
  - they report each student's grade on the exam, and his/her final course grade
  - the correlation of exam score to college grade contributes to AP score setting

    a score of 5 on the exam is meant to equate to *average* A-level performance in college
    a score of 4 on the exam is meant to equate to *average* B-level performance in college
    a score of 3 on the exam is meant to equate to *average* C-level performance in college
    a score of 2 on the exam is meant to equate to *average* D-level performance in college
    a score of 1 on the exam is meant to equate to F-level performance in college

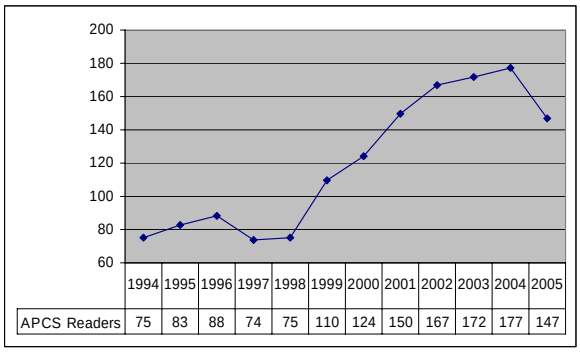
- in addition, some multiple choice questions are reused each year to allow for scoring consistency between exams

10

Grading ("Reading") Process

free response questions are graded by high school and college faculty in June at Clemson, SC

- 2005: 111 readers, 17 table leaders, 16 question leaders, 2 exam leaders, CR
- roughly a 45/55 college-to-high school ratio



	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005
APCS Readers	75	83	88	74	75	110	124	150	167	172	177	147

11

Grading Roles

the Chief Reader designs scoring standards (rubrics)

- questions are scored 0 to 9 (or – for "off task")

a few days before the reading, leaders examine sample responses and apply/refine the rubrics

- *question leaders* focus on refining/managing the rubric
- *table leaders* focus on applying the rubric and mentoring readers

readers (a.k.a. faculty consultants) arrive and grade the exams over the course of a week

- can specify preferences for which question they will grade

12

Exam Grading

readers receive extensive training on their problem and the scoring rubric in order to ensure consistent grading

- the reading starts with a short orientation
- leaders review the problem, canonical solutions, & rubric
- readers apply the rubric to "training packs" of pre-selected solutions, comparing results with question leaders
- readers then grade a "split pack" of 25 exams with a partner, comparing and discussing the results
- finally, they are ready to grade packs of exams on their own

13

Grading for Consistency

additional consistency checks are built into the process

- readers can consult with partners or table leaders for clarifications; even "star" difficult cases
- table and question leaders "backread" each pack of exams, reviewing starred questions and spot-checking others
- statistics on reader scoring tendencies are collected electronically and are reviewed by leaders
- periodically, reliability studies are conducted in which packs of exams are graded twice and scores correlated

Computer Science is consistently one of the most reliable and consistent grading subjects

14

Benefits for Participants

- community of computer science educators
- broad perspectives on CS1/CS2
- assessment experience and expertise
- professional development

to become a faculty consultant

- go to: http://www.ets.org/reader/ap
- or send an email to: apreader@ets.org
- or call: (609) 406-5384

15

Teacher Training

College Board provides training and professional development for high school teachers

- summer institutes
- regional workshops

training is conducted by endorsed consultants

sample syllabi and teacher resources are available online at AP Central
http://apcentral.collegeboard.com

16

A High School Teacher's Perspective

AP Test Development: committee process

- questions are typically written by one person
- then the committee takes over
- the finished product doesn't always look like the original product
- typical committee term is four years

grading the AP Exams

- must teach AP CS for 3 years before grading
- HS teachers can grade for 6 years at each level and then must retire for three years
- terrific experience

17

A High School Teacher's Perspective

College Board training for HS teachers

- one and two-Day workshops during the school year
- week-long summer institutes during the summer
- commitment level has to be high

teaching a college-level course to 14 – 18 year olds

- courses taught over a year
- lab time must be done during school day
- HS students not as mature – requires a HS teacher (me) to find ways to help them understand very abstract ideas

18

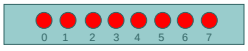
Observations from the AP CS Classroom

- Object Oriented Programming has added more abstraction to the AP Computer Science curriculum
- Object Oriented Programming has added a layer of complexity to the AP Computer Science curriculum
- Most of my students need concrete examples to help them understand the abstract concepts found in the AP Computer Science curriculum

19

Teaching Tips for AP Computer Science

- model algorithms before you show code
- use concrete models for abstract concepts
  - Bottle top array:



    - 8 bottle caps (20 oz soft drinks work best)
    - piece of cardboard
    - Hot glue the caps on the cardboard
    - Cut strips of card stock or index cards for the data
  - Towers of Hanoi toys for stacks
  - Beads on a string for queues (Toddler's pop beads for teacher demo)

20

Teaching Tips for AP Computer Science

- use concrete models for abstract concepts
  - create a “map” of students to crayon colors
    - 4 boxes of 8-color crayons
    - paper
    - hand each student a crayon.
    - build a map (on paper) of student names to crayon colors
    - some sample questions to ask:
      - How can you find which crayon Jack has?
      - How can you find all the students who have a specific color?

21

Teaching Tips for AP Computer Science

- use stories to introduce or reinforce concepts (e.g., Dr. Seuss)
  - to introduce arrays: "Too Many Daves" from The Sneetches and other stories
  - linked lists: "King Looie Katz" from I Can Lick 30 Tigers Today!
  - stacks: "Yertle the Turtle" from Yertle the Turtle and other stories
  - recursion: The Cat in the Hat Comes Back
    - also, "Martin and the Dragon Stories" from LISP:A Gentle Introduction to the Art of Symbolic Computation by David Touretzky
- use logic problems to foreshadow upcoming topics
  - great problems can be found in the following books:
    - The Puzzling Adventures of Doctor Ecco by Dennis Shasha
    - Problem Solving Strategies: Crossing the River with Dogs and Other Mathematical Adventures by Herr & Johnson

22

Teaching Tips for AP Computer Science

- software
  - Jeliot 3– Algorithm Theater - J*eliot 3 is a Program Visualization application. It visualizes how a Java program is interpreted by displaying method calls, variables, and operations, allowing the student to follow step by step the execution of a program.*
    http://www.cs.joensuu.fi/jeliot/index.php
  - Alice – Early OOP programming concepts without the overhead of a big language. www.alice.org
  - KarelJ Robot - Karel the Robot in Java – OOP programming without a lot of language overhead.
    http://csis.pace.edu/~bergin/KarelJava2ed/Karel++JavaEdition.html
  - BlueJ – Java IDE designed for teaching OOP early. http://www.bluej.org

23

FYI

- http://apcentral.collegeboard.com
  AP Central: AP info, course descriptions, reference materials, …
- http://apcentral.collegeboard.com/program/research
  AP Research and Data: exam data, research studies, …
- http://www.collegeboard.com
  College Board: general info about the association, AP program
- http://cs.colgate.edu/APCS/Java/APCSJavaMaterials.html
  Unofficial APCS site, by Chris Nevison (former Chief Reader)

24