**Chapter XX**

**The APCS 'A' Examination**

**Chapter XXII Topics**

20.1 Introduction

20.2 The Exam Format

20.3 The AP Java Subset

20.4 Java Standard Libraries

20.5 The Case Study

20.6 Multiple Choice Sample Questions

20.7 Free Response Sample Questions

20.8 Snooker Questions

20.9 Exam Day Important Reminders

**20.1 Introduction**

If you are in an AP Computer Science course then the arrival of this last chapter signals the coming of the AP Computer Science Examination in the very near future. If you have no knowledge of the topics that will be covered on the exam than this final chapter will do little to improve your score. On the other hand, even students with excellent knowledge and clear understanding of every exam topic can benefit from a summary of what to expect the second week of May.

This chapter is titled *The AP Computer Science 'A' Examination*. The College Board used to have two computer science examinations. The 'A' exam is based on the topics covered in a typical Computer Science I (first semester) college course. The 'AB' exam was based on the topics covered in the typical Computer Science II (second semester) college course. The 'AB' course no longer exists, but the remaining course is still designated as the 'A' course.

This chapter will look at the exam format and outline all the topics that may be presented, but something must be understood straight from the beginning. This exam is going to be tough. It is designed to be tough, because it is a college-level examination and this exam must convince universities that you should be awarded college credit for performing well. It turns out that performing well itself it a curious notion and high school students are so fixed on a system where 70% means a passing, but not very good, grade. You will be surprised that 70% is a very good performance on the AP Computer Science Examination. You need to forget the standard ideas about 90% is an 'A', 80% is a 'B', etc. That type of grading will be very discouraging during the taking of an AP exam. You may feel that you can only handle half the questions at best. Your normal high school experience will dictate that a 50% performance is failing. It may come as a surprise to you that scoring roughly 50% may well be a passing AP exam grade. Let us start by considering how AP examinations are reported.

|  |
| --- |
| **AP Examination Reported Grades** |
| **There are 5 different grades reported on AP Examinations.**  5 Extremely well qualified  4 Well qualified  3 Qualified  2 Possibly Qualified  1 No recommendation |

The five reported grades are not linked to a rigid *percentage-passing*, such as 80% equals 5 and 70% equals 4. You can use the following rough table to get an idea of what to expect. Keep in mind that this rough table is based on my experience as an AP Computer Science Exam reader and member of the committee who created the AP Computer Science Examination. Educational Testing Service (ETS), the organization which creates and administers AP Examinations for the College Board, administers very sophisticated statistics to arrive at the reported grade. The main goal of assigning grades is to insure that students of equal ability receive equal grades year-to-year on different tests.

|  |
| --- |
| **AP Computer Science Approximate Grade Scales** |
| 5 70% - 80%  4 60% - 70%  3 50% - 60%  2 40% - 50%  1 Less than 40% |

Now the point of this table is not at all for you to aim at some percentage. Frankly, you will not be any position to come close to accurately predicting what percentage of the test you answer correctly. The key thing is that you must realize not to get discouraged when you feel you are missing many questions. I mean by missing many questions your normal experience of getting 80% or better of the questions correct to make a decent grade. That simply is not the nature of an AP Examination.

This different percentage realization is very significant. I have known students who would shut down half way through an AP exam with the absolute conviction that any additional effort is a waste of time. Never make that assumption and do not quit when you feel that way. I have had students take AP Computer Science Examinations since the 1983-1984 school year and many, many students have earned 4s and 5s who were totally convinced immediately after the exam that they did poorly and did not expect to get a passing grade. Incidentally, most people consider a *passing grade* to mean **3** or better.

|  |
| --- |
| **The College Board Course Description** |
| The best guide to any AP exam is the *Course Description* published by the College Board. This guide is a detailed booklet with all the information published by the College Board to prepare teachers and students for the AP exam.  Teachers and students can both download the latest course description and lots of other AP information from the Internet.  Go to **http://apcentral.collegeboard.com/apc/Controller.jpf** |

**20.2 The Exam Format**

The world is full of experts and many offer lots of free advice. Be careful when fellow students tell you exactly what to expect on an AP Examination. After all these good friends already took an AP Examination and they know what is happening. There is a slight problem with their expertise. Knowledge of the format in one examination does not necessarily translate to knowledge in another examination. I will guarantee you that AP European History is very different from AP Computer Science. They both have a multiple choice section and a free response section, but then the similarities end. In many cases the times devoted to each section of the exam is not even the same length. My point? The exam format and the information provided in this chapter is specific for AP Computer Science. There is not some generic format that fits all AP exams.

You will take the three-hour exam in two parts. You start with a 40-question multiple choice test, followed by a free-response 4-question test. The multiple choice test is 75 minutes, followed by a break, and then you have 105 minutes for the free response questions. In many AP exams the free response section is called the *essay* test. This is precisely my point earlier. In English, History and many other subjects writing an essay is quite appropriate. It is a different story in Calculus, Physics and Computer Science.

|  |
| --- |
| **AP Computer Science Examination Length** |
| Part I is a multiple choice test.  There are 40 questions.  You have 75 minutes to complete part I.  Part II is a free response test.  There are 4 questions.  You have 105 minutes to complete part II. |

Later in this chapter you will see a variety of sample multiple choice questions and free response questions as well. This section is only meant to be an overview of the exam format that you will encounter. It may seem odd to be concerned about the exam format. After all, isn't the real focus on knowing the material that is being tested? Absolutely, and all the exam-format-knowledge in the world will not help you if you are clueless about the topics being tested.

However, there is another issue. Students, especially younger students, are very anxious about taking important tests. Anxiety about how many questions to expect, what type of questions will be presented and any other details distract from the focus of concentrating on answering the questions correctly. Any anxiety that can be reduced prior to the test is a plus, and that is the goal of this chapter. Frankly, if you have arrived at Chapter 20 and you do not understand the first 19 chapters, it may be wise to re-consider taking the examination at all.

|  |
| --- |
| **Multiple Choice Question Format** |
| Every multiple choice question has five answers.  There are 35 questions computer science topic questions.  There are 5 GridWorld Case Study questions. |
| **Free Response Question Format** |
| There are 3 computer science topic questions.  There is 1 GridWorld Case Study question.  Free response questions mostly require the completion of a specified method or the design of a class.  The majority of free response questions require the completion of multiple methods. |

**20.3 The AP Java Subset**

Sometimes it is difficult to realize that the title of the course you are taking states *Computer Science* and not *Java Computer Programming*. The first year course in Computer Science devotes most of its time writing computer programs. By the time you finish this Computer Science course you will start to realize that there are some topics that are not language specific and some topics are not computer programming at all. A good example you observed this year was the Boolean Algebra section. That very same section, with extremely small changes, was first used in *Student Friendly Pascal*, then *Exposure C++* and now it is used with *Exposure Java*.

This brings up a fascinating point. The AP Computer Science Examination presents programming code that needs to be analyzed and you are also expected to write program segments. These program segments are written in a specified language and you are expected to write code in the specified language. I was a *reader* - a person who grades AP Examinations - for many years and it amazed me that each year there were students who wrote perfectly correct solutions to free response questions in the wrong programming language. The correctness of such solutions is not a factor. There are no points awarded for using the wrong language. This fact is clearly stated on the examination booklet.

|  |
| --- |
| **The AP Computer Science Examination Uses Java** |
| *Answers to free response questions must be written in the specified programming language.*  *The current programming language is Java.*  *You will not receive any credit for a solution, no matter how correct, if the solution is written in any language other than Java.* |

It is simple to state that the programming language is Java, but what exactly does that mean. The total set of reserved words in Java is quite small and not so difficult to learn and test. On the other hand, there is an absolutely huge set of libraries available in Java with hundreds of classes and thousands of methods. You may remember that it was mentioned that Java provides around 60 classes just to handle input/output.

Teachers would go crazy if they had to prepare students for an examination that could cover any and all of the Java programming features. At the same time students would probably go on strike if they were shown the full extent of the Java program language features and libraries.

I mentioned earlier that introductory college-level computer science courses aim to teach fundamental programming concepts. The College Board has identified the most significant concepts that should be taught and additionally identified those concepts and features that will be tested. The College Board is not stating that an AP Computer Science course should be limited to these concepts. Rather it is stating that questions on the exam will be limited to those topics and Java features that are specified in the course description.

The College Board Course Description includes a detailed appendix on all the Java features that will be tested. The same appendix mentions items that may be relevant to teaching Computer Science, but will not be tested. I have listed a brief summary below of the items that will be tested. For complete details go to the official AP Computer Science Course Description.

|  |  |
| --- | --- |
| **Language Features** | |
| Simple data types | int  double  boolean |
| Integer class constants | Integer.MAX\_VALUE  Integer.MIN\_VALUE |
| Assignment operator | = |
| Arithmetic operators | + - \* / %  ++ --  += -= \*= /= %= |
| Logical operators and short circuiting | == != < <= > >=  && || ! |
| Type casting | (int)  (double) |
| String concatenation | + |
| Escape sequences | \" \\ \n |
| Text output | System.out.print  System.out.println |
| 1-dimensional arrays  2-dimensional arrays  initializer lists | list = new int[100];  matrix = new int[4][5];  int[] list = {1,2,3,4,5,6,7,8,9}; |
| Selection | if  if ... else |
| Repetition | while  for  enhanced <for..each>  return |
| Design a class  Modify a class  Inheritance  Composition | Design and implement subclasses  Modify a subclass  Conversion to supertypes and subtype casts  Extend classes |
| Class member access | public  private |
| Static class members | static final variables  static or class methods  initialization of static variables |
| implicit parameter | this |
| reference | null |
| access super class | super(args) to pass information to super class  super.someMethod(args) to access super class |
| Class constants | public static final MAX\_SCORE = 25; |
| Constructors | No-parameter constructors  Parameter - overloaded - constructors |
| Abstract class and interfaces | Design an interface  Implement an interface  Understand concept of abstract classes |
| Comparison | Difference between == != equals and compareTo |
| Package concept | Use import statements |
| Exceptions | Understand exception concept  Recognize common exceptions |
| Standard Java Library Classes | String  Math  Random  Object  List & ArrayList |
| Wrapper classes | Integer  Double |
| Generics | Generic Collection classes and interfaces  ArrayList class with generic type declaration |
| javadoc | @param, @return |
| Conversion | type casting and class casting (Subtype) |

**20.4 Java Standard Libraries**

It was mentioned earlier that Java has an extremely extensive library of interfaces and classes with an even larger set of methods. The AP Computer Science Test Development Committee has been kind and only a modest group of Java library methods will be tested on the 'A' exam.

Please keep in mind that the list shown below does not explain each one of these methods. For additional information you need to go to the appropriate chapter where these methods are introduced or go to the Java Documentation. You need to realize that you have learned a substantial amount of important computer science information that will not be tested. As you do final preparations for the upcoming examination, it is wise to now focus strictly on the items that will be tested. This chapter hopes to help you in narrowing your study focus. As you study the standard library methods below, keep in mind that the long class name includes the package names that contain the class. The big containing package is **java** and then you will note that most of the classes belong to the **lang** class or, to be more complete, the **java.lang** package. A few belong to **java.util**.

**Standard Java Libraries**

|  |
| --- |
| **class java.lang.Object** |
| **boolean equals (Object other)**  **String toString()** |

|  |
| --- |
| **class java.lang.Integer** |
| **Integer(int value)**  **int intValue()**  **Integer.MIN\_VALUE**  **Integer.MAX\_VALUE** |

|  |
| --- |
| **class java.lang.Double** |
| **Double(double value)**  **double doubleValue()** |

|  |
| --- |
| **class java.lang.String** |
| **int length()**  **String substring(int start, int end)**  // returns the substring from ***start***to ***end-1***  **String substring(int start)**  // returns substring from ***start***to ***length()-1***  **String indexOf(String str)**  // returns index of first occurrence of **str** andreturns -1 if not found  **int compareTo(String other)**  // return value < 0 if **this** is less than **other**  // return value = 0 if **this** is equal to **other**  // return value > 0 if **this** is greater than **other** |

|  |
| --- |
| **class java.lang.Math** |
| **static int abs(int x)**  **static double abs(double x)**  **static double pow(double base, double exponent)**  **static double sqrt(double x)**  **static double random()**  // returns a double in the range [0.0 ... 1.0] |

|  |
| --- |
| **interface** **java.util.List<E>** |
| **int size()**  **boolean add(E obj)**  // appends **obj** to end of list; returns **true**  **void add(int index, E obj)**  // inserts **obj** at position **index** (0 <= index <= size)  // moving elements at position index and higher  // to the right (adds 1 to their indices) and adjusts size  **E get(int index)**  **E set(int index, E obj)**  // replaces the element at position **index** with **obj**  // returns the element formerly at the specified position  **E remove(int index)**  // removes element from position index, moving elements  // at position **index+1** and higher to the  // left (subtracts 1 from their indices) and adjusts size  // returns the element formerly at the specified position |
| **class** **java.util.ArrayList<E> implements java.util.List<E>** |
|  |

**20.5 The Case Study**

It was mentioned earlier that a Case Study is tested and five multiple choice questions along with one free response question are devoted to this topic. Now let us do a little math. Each section, multiple choice and free response, is 50% of the total exam grade. In the multiple choice section 5 questions out of 40 equals 1/16 of the total test. In free response 1 question out of 4 equals 1/8 of the total test. Combined this means that 3/16 or 19% of the entire examination is devoted to the current GridWorld Case Study.

|  |
| --- |
| **Case Study Exam Significance** |
| Questions about the Case Study on the AP Computer Science Examination account for **3/16** or **19%** of the total exam grade. |

I have had my share of grading AP Computer Science Examinations and the Case Study is *feast or famine*. If you have studied and worked with the Case Study then you will find the Case Study questions very comfortable. On the other hand, if you think that the GridWorld Case Study is an effort by urban planners to reduce traffic congestion then you will be in for some perplexing questions.

In brief, the Case Study is a large program designed and implemented by a group of master programmers. This large program is presented to introductory computer science students in an effort to learn about proper program design and to learn how to alter and enhance an existing program.

Where is there information on the current Case Study? The College Board website provides many tools and links to help teachers with the case study. Since 2012 the GridWorld Case Study is integrated in the Exposure Java textbook for AP Computer Science.

This section will make no attempt to teach the details of this case study. Hopefully, you have learned about this case study throughout the year or perhaps you have already studied a concentrated unit for two or three weeks. It is also possible that you are looking at this chapter with sufficient time left to familiarize yourself with this important program. If you know nothing about the Case Study and there is a little time left, get busy in a hurry. If the AP Computer Science Examination is tomorrow then I have bad news for you. It is too late.

**20.6 Multiple Choice Sample Questions**

You will start with the multiple choice questions and most students find this first section easier than the later free response section. Be careful that you do not pick the "correct" answer too quickly. I learned a few things during the four-year period that I served on the Test Development Committee. A large amount of time is devoted on writing these questions and an even more amazing amount of time is used to create plausible distracters. A *distracter* is a wrong answer and each question has four wrong answers. Great care is taken to make the distracters seem reasonable. Forget looking for some off-the-wall nonsense answer that can be eliminated instantly. Each distracter is selected with a purpose.

AP exams used to penalize guessing. Every wrong answer on the multiple choice section resulted in a 1/4 point deduction. This is no longer the case.

|  |
| --- |
| **There is no longer any penalty for guessing the wrong answer on the multiple choice section of the AP Computer Science Examination.** |

You must realize that the multiple choice test does not ask questions based on the memorization of facts. There are no questions that ask something like *What Java reserved words are used in a two-way selection control structure?* The questions are designed to see if you have more than superficial knowledge. Do you really understand the computer science concepts well enough to analyze a question correctly and then select the correct answer. The questions are hard, but there is the good news. These questions are not just given to you; they are given to all the other students as well.

Examine the four questions that follow. They are written at the difficulty level that you might expect on the AP Computer Science 'A' Examination. This does not cover all the topics. If you want a more thorough set of questions, you can find them in the College Board Course Description and a large selection of AP study guides books are published to help you prepare for the AP Examination.

|  |
| --- |
| 01. The Boolean expression  (A && B) && !(A || B)    evaluates to  (A) true in all cases.  (B) false in all cases.  (C) true, whenever both a is true and also b is true.  (D) true, whenever either a is true or b is true.  (E) false only, whenever either a is true or b is true. |

|  |
| --- |
| 02. Consider the following method.  public static int mystery(int m)  // precondition: m > 0  {  int counter = 0;  for (int p = 1; p < m; p++)  for (int q = m; q > 1; q--)  counter++;  return counter;  }  What value is returned as a result of the call mystery(x) ?  (A) (x - 1)2  (B) x2  (C) x2 - 1  (D) x2 - 2  (E) (x - 2)2 |
| 03. Consider the following method.  public static void mystery(int[] list)  {  int n = list.length - 1;  for (int k = 0; k < n; k++)  list[k] = list[n - k];  }  Assume that an integer array contains the values: {11, 45, 32, 65, 47, 21, 89, 50, 39}  Which of the following represents the contents of list as a result of calling method mystery ?  (A) {39, 50, 89, 21, 47, 65, 32, 45, 11}  (B) {11, 45, 32, 65, 47, 65, 32, 45, 11}  (C) {39, 50, 89, 21, 47, 21, 89, 50, 39}  (D) {11, 45, 32, 65, 47, 11, 45, 32, 65}  (E) {21, 89, 50, 39, 47, 11, 45, 32, 65} |
| 04. Consider the following method.  public static String weird (String str)  {  String temp = "";  for (int k = 0; k < str.length(); k++)  {  if (k % 3 == 0)  temp += str.substring(k,k+1);  else  temp += "X";  }  return temp;  }    What value is returned as a result of the call weird("COMPUTER") ?  (A) XOMXUTXR  (B) XXMXXTXX  (C) COXMPXTE  (D) CXMXUXEX  (E) CXXPXXEX |

**20.7 Free Response Sample Questions**

The free response questions can be intimidating. At first glance it is very natural to think that you can never do this stuff. Let it pass. Take a deep breath and start to dig into the first part of the first question. You will see two sample questions shortly, and if you have never seen a sample free response question, the examples that follow give you an idea of the format.

Technically there are four free response questions, but many questions are divided up into multiple parts. Keep in mind that it is not necessary to understand or even read part (b) and part (c) when you get started with part (a). The anxiety will start to disappear when you get started and you will realize that the question is not all that bad.

The first mission is to comprehend clearly what is expected. The majority of questions require that you complete one or more methods. The different parts normally work in sequence, which means that the method completed in part (a) can be used to write the method in part (b). Do not fall into the trap of trying to do everything at once. Do part (a) of the first question and when you only look at one part of one question, the question is less intimidating.

Do not be too concerned if you make some small Java syntax error. The test readers look for substance and logical correctness. Few points are lost by making some trivial mistake like leaving out a semi-colon at the end of a statement.

Free Response Question 1.

There are three measures of central tendency in statistics. One of the measures of central tendency is the arithmetic mean, which is calculated by adding all of the numbers and dividing by the quantity of the numbers. A second measure of central tendency is the median, which is the middle value of a list of n ordered numbers. If n is even, the median is the mean of the two middle numbers.

Part (a).

Write method createList, which constructs an ArrayList object, generates a set of random integers in the **[10..99]** range, stores these integers in the ArrayList object, and returns the array object.

Complete method createList below.

/\*\*

\* **Precondition**: n > 0

\* **Postcondition:** returns an ArrayList object with n Integer values;

\* each Integer object is created with a random int in the [10..99] range

\*/

public static ArrayList createList (int n)

Part (b).

Write method getMean, which is described as follows. Method getMean returns the average, called the **mean**,of a set of numbers. The **mean** is computed by adding the numbers and dividing the sum by the quantity of the numbers.

For example:

The **mean** of {10, 18, 17} is **15**

The **mean** of {10, 15} is **12.5**

Complete method getMean below.

/\*\*

\* **Precondition:** list is a non-empty array with Integer objects

\* **Postcondition:** returns the mean of the int values stored by the list array.

\*/

public static double getMean (ArrayList list)

Part (c).

Write method getMedian, which is described as follows. Method getMedian returns a measure, called the median,of a set of **n** numbers. The median is the middle number in a sorted list of numbers. If **n** is even, the median is the mean of the two middle numbers.

For example:

The **median** of {10, 13, 17, 18, 21} is **17**.

The **median** of {54, 60, 64, 68, 74, 81, 94} is **68**

The **median** of {10, 11, 13, 17, 19, 21} is **15**.

The **median** of {50, 60, 70, 80} is **65**

In writing method getMedian you may call method sortList specified below.

/\*\*

\* **Precondition:** list is a non-empty ArrayList object of Integer elements

\* **Postcondition:** The Integer values in list are sorted in ascending order

\*/

public static void sortList (ArrayList list)

Complete method getMedian below.

/\*\*

\* **Precondition:** list is a non-empty ArrayList object of Integer elements

\* The Integer elements in list are randomly ordered

\* **Postcondition:** returns the median value of list

public static double getMedian (ArrayList list)

Free Response Question 2.

This question involves identifying palindromes. A *palindrome* is a string of characters that reads the same from front-to-back and back-to-front. Three examples of palindromes are **racecar**, **madam** and **123454321**.

These first three examples are perfect palindromes. It is possible to read the string of characters from either end with the exact same results, without any special conditions. There are other strings of characters, which can be considered palindromes, but they are not perfect. Three examples of imperfect palindromes are **Racecar**, **Madam I'm Adam** and **A man, a plan, a canal, Panama**. **Racecar** can be considered a palindrome if case-sensitivity is ignored. The other two examples are also palindromes if case-sensitivity, spaces and punctuation are ignored.

For this question a palindrome shall be defined as a *string of alpha-numeric characters that reads the same from front-to-back and back-to-front without case-sensitivity*. This definition means that strings like **Racecar** and **Madam** will be considered palindromes, but **Madam I'm Adam** is not a palindrome.

Consider the following incomplete declaration of a Palindrome class. The constructor determines if the parameter is a palindrome by using the isPalindrome method. The private method isPalindrome needs to return true if the string is a palindrome and false otherwise. The method toString needs to be redefined to produce the output shown in the execution example.

class Palindrome

{

private boolean palindrome;

private String str;

Palindrome(String s)

{

str = s;

palindrome = isPalindrome();

}

public String toString()

{

/\* to be redefined in part (a) \*/

}

private boolean isPalindrome()

{

/\* to be implemented in part (b) \*/

}

}

The code segment below is part of a client program that constructs and displays two Palindrome objects.

|  |  |
| --- | --- |
| **Code segment in client program using the Palindrome class** | **Execution Output** |
| Palindrome p1 = new Palindrome("Racecar");  System.out.println(p1);  Palindrome p2 = new Palindrome("Madam I'm Adam");  System.out.println(p2); | String: Racecar  Palindrome: true  String: Madam I'm Adam  Palindrome: false |

Part (a).

Redefine the toString method as discussed earlier. Method toString must return a string that will display two lines of output. The following table shows the result of several different calls made by System.out.println(/\* Palindrome Object \*/) based on the argument of the constructor.

|  |  |
| --- | --- |
| **Constructor argument** | **toString returns** |
| Racecar | String: Racecar  Palindrome: true |
| 12345BOB54321 | String: 12345BOB54321  Palindrome: true |
| Madam I'm Adam | String: Madam I'm Adam  Palindrome: false |
| MADAM RACECAR | String: MADAM RACECAR  Palindrome: false |

Complete method toString below.

**/\*\***

**\* precondition:** str is not null

\* palindrome is true or false

\* **Postcondition:** returns a string in the format:

\* String: value of str

\* Palindrome: value of palindrome

\*/

public String toString()

Part (b).

Write method isPalindrome as described earlier. In writing method isPalindrome you may call method toUpperCase specified below.

/\*\*

\* **Precondition:** some String object s is constructed

\* **Postcondition:** returns s with all lower-case characters converted to upper-case characters,

\* if they exist

\*/

public String toUpperCase()

Complete method isPalindrome below.

/\*\*

\* **Precondition**: str is not null

\* **Postcondition**: returns true if str is a palindrome, and false otherwise

\*/

private boolean isPalindrome()

One of the reasons that free response questions are so intimidating is the quantity of reading involved. It looks overwhelming at first, but start reading. Do not read too much. Too much reading results in losing track of what you are doing and you may need to re-read the first part again. Read part (a) once and possibly twice and then get to work answering part (a). As you continue on to part (b) make sure that you realize that each part is graded separately. Your part (a) can be totally wrong or totally missing. No problem. You can still do part (b) and the score on part (b) is completely independent from your performance on part (a).

**20.8 Snooker Questions**

One of the most frustrating aspects of being an AP Computer Science teachers is that frequently very bright students, with excellent understanding of all the curriculum topics, fail the APCS Exam. In the last five years I have found that such students often are *snookered*. They look at a question and conclude that the topic is easy, the question is not very challenging, and it really only requires minimal brain effort.

I have a message for all the super bright students. There exist *snooker questions* that look deceptively easy and you miss some important detail. In this section I will present you with two questions. I am telling you up front that in my experience more than 75% of the students get these questions wrong. You have an advantage right now, because you are warned that there is a large change that you will get the question wrong by not paying attention.

Do not look for the answers somewhere in this chapter. You will not find it. You should also consider that on average you have less than two minutes per multiple choice question. If you read this at home and spent 30 minutes on each question, do not come in with an attitude that the question was not so tricky. You need to consider that students are anxious, because of the test and they are under pressure since there is little time. In such an environment it is easy to miss something.

My point in presenting this section is not to discourage you. The AP Computer Science Examination is not a set of 40 snooker questions. My point is to wake up the students who are very good with computers and they are masters in using the computer to debug their programs. Well, news flash. There are no computers during the AP exam. This is simply a warning to settle down and do not underestimate a question. You could be *snookered*.

|  |
| --- |
| Snooker Question #1  **01.** **Consider the following code segment.**    **int[] list = {11,22,33,44,55,66,77,88,99};**  **for (int k = 0; k < list.length; k++)**  **list[k] = list[k]/list[0];**  **for (int k = 0; k < list.length; k++)**  **System.out.print(list[k] + " ");**    **What is printed as a result of executing the code segment?**  **(A) 11 22 33 44 55 66 77 88 99**  **(B) 1 2 3 4 5 6 7 8 9**  **(C) 1 1 1 1 1 1 1 1 1**  **(D) 1 22 33 44 55 66 77 88 99**  **(E) 11 22 33 44 55 66 77 88 9** |

|  |
| --- |
| Snooker Question #2  **02. Consider the following MysteryCritter class.**  **class MysteryCritter extends Critter**  **{**  **public void turn()**  **{**  **int add = (int)((Math.random()\*90)+90);**  **setDirection(getDirection() + add);**  **}**  **}**  **How many degrees will a MysteryCritter object turn during one step of a**  **GridWorld execution?**  **(A) 0**  **(B) 179**  **(C) 180**  **(D) A random number in the range of 90 – 180**  **(E) A random number in the range of 90 – 179** |

**20.9 Exam Day Important Reminders**

We now come to a very important section in this chapter. It is the day of the AP Computer Science Examination. It is also too late to learn anything that you should have learned earlier. You know what you know and your knowledge is hopefully sufficient to do well on the exam. The question now becomes, what can you do to help or hurt your performance on the exam. Please read this section over very carefully. I have been involved with AP Computer Science since 1983, which was its very first year and I have graded these exams for many years. Student with perfectly good knowledge can lower their score by doing certain dumb things that are easily avoidable. This section will be divided up into the two exam sections for multiple choice and for free response.

**Multiple Choice Section Day-Of-Exam Hints**

Do not be concerned about deciding between doing multiple choice or free response first. The multiple choice test will be given to you first. It is already decided and a done deal. Great, this is one less decision you need to make.

We must start with some basic math. The multiple choice test has 40 questions and you have 75 minutes to finish this part of the exam. This means you have less than two minutes for each question. This simple reality means that finishing the test becomes an issue and time cannot be wasted. Each one of the 40 questions carries the same weight and some questions are more time consuming. Some questions are time consuming, because it involves a lot of reading. Other questions are time consuming, because you need to do of lot of calculation, tracing or some processing to derive at the correct answer. This means that you need to skip a question if you feel it is time-consuming. Come back to it later if you have the chance.

|  |
| --- |
| **AP Examination Multiple Choice Hint #1** |
| You have 75 minutes to complete 40 questions. Completion time is a definite factor. You have less than two minutes per question.  Skip time consuming questions and return to them later. |

Guessing used to be discouraged on AP exams and it was penalized as well. There *used to be* a 1/4 point deduction for every wrong multiple choice question. This is no longer true and there is no penalty for guessing.

Go ahead and guess, but mark the question and guess near the end. Guessing should be at least a process where certain answers are eliminated. If you must, guess out of five answers, but reduce the probability as much as possible.

|  |
| --- |
| **AP Examination Multiple Choice Hint #2** |
| Wrong multiple choice question answers are ***NO LONGER PENALIZED*** with a ¼ point "guessing" deduction.  If you are not sure, mark the question. Return later and then guess by eliminating as many answers first as possible. |

Students make careless mistakes during test taking. You may get "tricked" by a clever wrong answer or you may do something quite silly. That is fine. Read the question, determine the correct answer and then check to make sure that your logic, computation and details are correct. Make sure that you do not fall into the ***Snooker Trap***.

|  |
| --- |
| **AP Examination Multiple Choice Hint #3** |
| Check a question immediately after you have determined the correct answer and **do not get snookered**. There is not adequate time to thoroughly recheck questions after you are finished. |

Be aware that you will go through two very different test-taking modes. Initially, you look at a question and you are in your *reasonable* mode. You look at a question and use your knowledge and logic quite well. Now you are finished and then you switch to your *paranoid* mode. In such a case you may be inclined to start playing peculiar psychological games with yourself. You already answered the question correctly. Now you start doubting yourself. Was the answer too easy? Did you miss the trick that was planned to trap you? Many studies have concluded that students who change multiple choice answers make more changes from correct to wrong then wrong to correct.

|  |
| --- |
| **AP Examination Multiple Choice Hint #4** |
| When you are finished and you have answered every question, do not make any changes.  At this stage you may become paranoid and studies have shown that students change more questions from correct to wrong than wrong to correct. |

Many questions become involved. This is especially true for a question where you must determine the value of some variable at the conclusion of a program segment. The program usually includes a variety of complex control structures. These questions should be done on paper. It is very easy to get confused in your mind. Use a *variable trace* to determine values as the program execution proceeds. At the AP exam you are not allowed to bring or use scratch paper, but you can write directly on the question booklet, which has ample scratch space.

|  |
| --- |
| **AP Examination Multiple Choice Hint #5** |
| Use a variable trace to determine the value of variables when complex control structures are used.  You may use the test booklet as scratch paper. |

**Free Response Section Day-Of-Exam Hints**

Time is a factor for the multiple choice section, but the free response section has even greater time pressure. You do get 105 minutes, but in that time you need to read the questions, determine what is needed, configure solutions and write the answers. This is a time consuming process.

The next hint is so often ignored by students and it is a shame. Students probably learned during the entire year that it is important to make meaningful comments. This is an excellent habit that will assist in debugging and enhancing a program. However, it is a big waste of time on the AP Computer Science Examination. The readers - exam graders - of the exam follow a very precise set of rules to assign points for everything that you do correctly. Points are never added for useful, correct comments and points are never deducted for missing or wrong comments. Comments take a considerable amount of time. Time and again I have observed free response test booklets with extensive comments for the first three questions and then it is clear the student never made it to the last question.

|  |
| --- |
| **AP Examination Free Response Hint #6** |
| Do not write any comments with any of your answers.  Comments cannot add points, but they can lose points indirectly if you do not have sufficient time to finish all the questions. |

Write neatly!!!!! After many years of teaching I have found male students especially go through a hand writing deterioration phase somewhere between the 9th grade and the 12th grade. Students with sloppy, impossible-to-read hand writing, use the pitiful excuse that they can read their own writing. Well, let me congratulate you on being able to decipher your own hieroglyphics. There is just one little detail: You are not grading the test; a total stranger is grading the test and if the stranger cannot read your writing, you will not get points. I have many times felt that a student probably had a correct solution, but the scribbles were so horribly written that I could not give credit. Yes, you are in a hurry, but not in such a hurry that everything you write is a waste of time, because nobody can determine what you have written.

|  |
| --- |
| **AP Examination Free Response Hint #7** |
| Write neatly. Better yet, print neatly.  Exam graders - readers - cannot assign points to any code that is not legible. |

The next hint should not be necessary, but sadly, experience has shown that some students simply are unaware. All free response answers must use the language Java. The exam booklet will state that Java must be used, but there are always some students who seem to think that using some other language is fine. Well it is not and using some other language is a waste of time and exam money.

|  |
| --- |
| **AP Examination Free Response Hint #8** |
| Write your free response program segments in Java.  All other languages are ignored and will not earn points, even if the solution is written correctly in some other language. |

Free response questions have multiple parts. Each part of the question is graded separately. Students sometimes give up on a question when part(a) is too difficult. It may surprise you that many times part(b) is simpler or at least for you it may be simpler.

|  |
| --- |
| **AP Examination Free Response Hint #9** |
| Check out every part of a question. Do not stop if you cannot do part (a). Credit is given for each completed part of the question. |

Most free response questions are written in such a sequence that the method completed for part(a) may be used in the solution for part(b). Now the beauty is that you may call the method created in part(a) regardless of what you did for part(a). Perhaps your solution is wrong or possibly you did nothing for part(a). It does not matter. Go ahead and use the method discussed in part(a) as if a fully correct version exists.

|  |
| --- |
| **AP Examination Free Response Hint #10** |
| When it is suggested to use a method in part(a), regardless of the correctness of that method, makes sure to use that method call. It makes part(b) simpler and shorter. |

Each solution has components, such as: is a variable properly initialized? Is a loop entered correctly? Is a loop exited correctly? Each one of these components is awarded points. This means that you may realize for a certain method that you must initialize a variable and that a looping process must occur. You may be clueless what happens inside the loop. Ironically, it may be that 1 point is earned for initializing the variable and a second point is earned for looping properly. The entire method may be 5 points, but 2 points is better than 0 points.

|  |
| --- |
| **AP Examination Free Response Hint #11** |
| Write what you can. You may earn a few points for a partial solution. Some points are better than no points. |

Finally, do not sweat the small stuff. You will not lose points, or at worst not many points for small syntax mistakes. Forgetting a semi-colon after a statement or misspelling a variable is not important and usually ignored. Normally deductions will only kick in with a pattern, such as no semi-colons anywhere.

|  |
| --- |
| **AP Examination Multiple Choice Hint #12** |
| Do not be concerned about minor syntax errors. Infrequent syntax errors have either no deductions or very small deductions. |