

Fall Semester Study Guide

Monday, December 19, 2016 9:36 AM

AP Computer Science First Semester Study Guide

Base Conversions

1. Decimal (base 10) Use the divide by base system to convert)
2. Binary (base 2)
3. Octal (base 8)
 - a. Convert to Binary is a 3 to 1 relationship
 - b. Convert to Hex is the two-step process (convert to binary then to Hex)
4. Hexadecimal (base 16)
 - a. Convert to Binary is a 4 to 1 relationship
 - b. Convert to Octal is the two-step process (convert to binary then to Octal)
5. Remember...if you have to do addition or subtraction, convert to decimal first.

Chapter 1

1. Reserved words – special purpose (all lowercase)
2. White space – ignored by computer, used just so humans can read the code
3. Comments – used to explain the code
 - a. Line comments use //
 - b. Multi-line comments use /* at beginning and */ at the end
4. Identifier rules – identifiers are names for variables, methods, and classes
 - a. Can't start with a digit (0 – 9)
 - b. Can only use \$ and _ characters
 - c. Classes use a uppercase first letter
 - d. Constants are all uppercase
 - e. Use title case (qtyOnHand)
5. Errors
 - a. Compile errors – syntax errors
 - b. Run-time errors – occur at run time
 - i. Best example is a divide by zero error
 - c. Logic errors – most difficult to detect and fix. You have to test your results to discover and correct.
6. Program development – know the steps. Understand that the step which is implementation (coding) is the least creative.

Your Notes

AP Computer Science

First Semester Study Guide

Chapter 2

1. Now the escape sequences (must start with a \)
 - a. Understand they have to be inside a string literal
2. Data types (objects & primitive)
3. Primitive types include:
 - a. int – integers
 - b. double – values with a decimal point
 - c. booleans – valid values are only true and false
 - d. char – character literal (must be inside single quotation marks 'a')
4. There are three types of data conversions
 - a. **Assignment conversion**: (double num = 3; a widening occurs)
 - b. **Arithmetic promotion**: (5 / 3.0; since one of the operands is a double, the answer is rendered as a double. Again, a widening conversion occurs)
 - c. **Casting** is when you put type you want to apply in parentheses. This can be done with widening and narrowing so it is powerful and dangerous!
 - d. Know the difference between narrowing and widening conversions
5. The methods of the String class:
 - a. Know all the methods that are listed on page 84
 - b. substring()
 - c. length()
 - d. indexOf()
 - e. compareTo()
 - f. equals()
6. The methods of the Math class:
 - a. Know that these methods are **static** – meaning that you call them by using the class name and not an object
 - b. **random()** method – please know how to use the random method to pull any type of random number in any range. See your worksheets if you need to refresh.
7. Wrapper class
8. Know what a **wrapper class** is and why it is used.
 - a. Due to sometimes we need to treat a primitive type as an object
 - b. Contain some valuable constants that we can reference like:
 - i. Integer.MIN_VALUE
 - ii. Integer.MAX_VALUE
9. Know your order of operations!
10. Understand that class files are organized into packages
 - a. Java.lang is the main package where the String, System, Math, and other commonly used classes reside. We don't need to import these if we want to use them
 - b. Other classes like the Scanner class is part of the utility package and we would have to import it in order to use it: `import java.util.Scanner;`

AP Computer Science

First Semester Study Guide

Chapter 3

1. Know the correct syntax for if statements, else-if, nested if's
 2. know the Boolean operators
 - a. && logical and
 - b. || logical or
 - c. ! logical not
 - d. == equality operator
 - e. != not equals to
 3. Know DeMorgan's Law and how to distribute a ! sign.
 4. Know the proper syntax for a while loop. Remember that you have to perform the initialization prior to the loop
 - a. a sentinel value allows the user to stop the loop
 - b. We use a while loop when the programmer doesn't know in advance how many times the loop will iterate
 5. Know the proper syntax for a for loop.
 - a. Please understand that the increment or decrement occurs at the end of the loop cycle
 - b. We use a for loop when the programmer knows in advance how many times we want the loop to iterate
 6. Know how to set up truth tables and the algorithm for how many scenarios you have based on the number of variables
 - a. two variables equals 4 scenarios
 - b. three variable equal 8 scenarios
 7. Understand short-circuiting
 - a. when the left side of the equation gives us enough information that we don't have to check the right side
 - b. the left side is false in a && statement
 - c. the left side is true in a || statement
 - d. Great to use to prevent a divide by zero run-time error
- Know how to trap the max and min values
- e. know how to instantiate correctly
 - f. know how to use the if statement to trap the values
- 8.

AP Computer Science

First Semester Study Guide

Chapter 4

1. Know the four steps to write a class file
 - a. class header
 - b. declare the private instance variables
 - c. write the constructors
 - d. write the other accessor and mutator methods
2. Know about the **scope** of the declared variables
 - a. **local** variables are declared inside a method and can only be referenced inside that method
 - b. **class level** variables are declared with the other instance variables and can be referenced from anywhere in the class file
3. Understand the concept of **encapsulation** (to protect the integrity of the instance variables)
 - a. all instance variables should be made private. If you leave off the visibility modifier, it will default to public
 - b. For the clear majority of the time, methods should be made public to provide functionality for the clients. Clients can call the public methods from a runner program.
 - c. Private methods are only used as a helper or support method that will be only called from inside the class.
4. Know the method header of the toString method and why it is used
 - a. It comes from the object class and so it is overridden in all class files
 - b. the header is: public String toString()
 - c. You must have a return statement in the method that returns a string
 - d. The method is called automatically when you try to print an object
 - e. If the objects class file does not have a toString method, it prints out the hexadecimal location of the data's memory address. Hexadecimal "**gobbledygook**"
5. Know the difference between the formal parameters and actual parameters of a method
 - a. formal parameters of a method are local variables
6. Know that the reserved word "**this**" refers to the object calling the method or the instance variable of the class file.
7. Know the rules for **constructors**
 - a. They have the same name as the class
 - b. They are always public
 - c. They have no return type (not even void)
 - d. They are often overloaded meaning: the signature of the parameters are unique.
 - e. The signature is the number, type, and order of the parameters

Chapter 6

1. Know that arrays in Java are objects.
2. Know the proper way to instantiate:
 - a. `int[] num = new int[10];`

AP Computer Science

First Semester Study Guide

3. Know the proper way to use an initializer list. The portion to the left of the = is the same. To the right of the =, just list the contents between {} separated by commas.
 - a. `int[] num = {3,5,2,76,32,45,12,1};`
 - b. Make sure char's and string's use the proper syntax: 'F' or "tom"
4. The constant `length` is created when the array is instantiated
5. The for loop to use to traverse an array is:
 - a. `for (int i = 0; i < array.length; i++)`
6. Know that when you instantiate an array the value are set to:
 - a. Zero for doubles and ints
 - b. False for Boolean
 - c. Null for objects
7. Know the three line swap algorithm
 - a. `int temp = nums[0];`
 - b. `nums[0] = nums[5]`
 - c. `nums[5] = temp;`
8. Know the algorithm for reversing an array
 - a. `for (int i = 0; i < array length/2; i++)`
 - b. `{`
 - c. `int temp = array[i];`
 - d. `array[i] = array[array.length-1-i];`
 - e. `array[array.length-1-i] = temp;`
 - f. `}`