* Golub
  + 1.  What does the acronym CVS stand for?
    - Concurrent Versioning System
  + 2.  What does IDE stand for?
    - Integrated Development Environment
  + 3.  What IDE will we use in class this semester?
    - Eclipse
  + 4. What does CPU stand for?
    - Central Processing Unit
  + 5. Define the term "bit".
    - A binary value
  + 6. Define the term "byte".
    - 8 bits
  + 7. What is "hardware"?  Name some examples.
    - The physical part of the computer. e.g. motherboard, cpu, ram
  + 8. What is "software"?  Name some examples.
    - The programs e.g. OS, Microsoft word
  + 9. Name several different operating systems.
    - Linux, Unix, Windows, OSX
  + 10. Give some examples of secondary memory devices.
    - USB, CD, Hard disk,
  + 11. What is the advantage of primary memory over secondary?
    - Faster
  + 12. What is the advantage of secondary memory over primary?
    - Permanent
  + 13. What does I/O stand for?  Give some examples of I/O devices.
    - Input/Output – keyboard, mouse, turnoff button
  + 14. How many different combinations of 0's and 1's can be represented using 7 bits?
    - 128
  + 15. How many bytes are in a kilobyte?  Megabyte?  Gigabyte?
    - 1000/1024 – kilobyte | 10^6 – Megabyte | 10^9 – Gigabyte
  + 16.  Name four things that the operating system does for you.
    - I/O
    - Manages passwords and files
    - Manages memory
    - Process Management
  + 17.  What do you call the language that the CPU uses (0's and 1's represent instructions in this language).
    - Machine Code
  + 18.  How does "assembly language" relate to your answer to the previous question?
    - Allowed names to be given to addresses
  + 19.  Name some higher level languages that were NOT designed for object oriented programming.
    - C, Cobol, Fortran
  + 20.  Name some higher level languages that WERE designed for object oriented programming.
    - Java, C#, C++
  + 21. Translate the number 123 into base 7 representation.
    - 234
  + 22. Translate the binary (base 2) number 1011010 into base 10 representation.
    - 64+16+8+2 = 90
  + 23. Translate the hexidecimal (base 16) number 7F into binary representation.
    - 01111111
  + 24.  When your Java program is compiled, what type of file is created?  (Hint:  It is NOT machine language.)
    - bytecode
  + 25.  What does it mean for someone to say that a Java program is "portable"?
    - Can run any machine with JVM
* Golub
  + 1.  When your Java program is compiled, what type of file is created?  (Hint:  It is NOT machine language.)
    - bytecode
  + 2.  What does it mean for someone to say that a Java program is "portable"?
    - The program can run on any machine with JVM
  + 3.  TRUE/FALSE:  Inserting unnecessary spaces and/or blank lines could cause your Java program to malfunction.
    - False
  + 4.   What is the difference between "syntax errors" and "logical errors"?
    - syntax error are typos – logical errors are errors that run but don’t give you the right result
  + 5.   If your program compiles and runs, but behaves incorrectly, are you probably suffering from "syntax" or "logical" errors?
    - logical
  + 6.  If Eclipse flags your code with a red mark and won't let you compile it, are you suffering from "syntax" or "logical" errors?
    - syntax
  + 7.   List the four Java primitive types that can be used to store integer values.
    - byte
    - short
    - int
    - long
  + 8.   How much memory is required to store each of the four types mentioned in the previous question?
    - 1
    - 2
    - 4
    - 8
  + 9.  What advantage do you gain from using one of the types of integer types that requires MORE memory?
    - Wider range of numbers
  + 10.   List the two Java primitive types that can be used to store floating point values.
    - Float
    - double
  + 11.   How much memory is required to store each of the two types mentioned in the previous question?
    - 4 bytes
    - 8 bytes
  + 12.  What advantage do you gain from using one of the floating point types that requires MORE memory?
    - Real numbers
  + 13.  List the two Java types that are used to store values that are not numbers.
    - Char
    - Boolean
  + 14.  Write a statement that declares a variable named counter of type int, and stores the value 182 in the variable.
    - Int counter =182
  + 15.   Write a statement that simultaneously declares three variables of type boolean, named x, y, and z.
    - Boolean x, y, z;
  + 16.  What values can a boolean variable achieve?
    - True or false
  + 17.  Write a java class called "Fred".  Put in a main method.  Have the main method store your age in a variable named age. Then main should print out a line that has your name, followed by your age.  (Use a "string literal" for your name, but use the variable to access your age.)
    - public class Fred {

public static void main(String[] args) {

int age = 22;

String name = “Chinedum”

System.out.println(name+“ ”+age);

}

}

* + 18.   Practice using BOTH styles for Java comments.
  + 19.   Is the following statement valid:     int x = 34.7;
    - NO
  + 20.   Is the following statement valid:     double y = 12;
    - YES
  + 21.   Is the following statement valid:      boolean q = 17 < 25;
    - YES
  + 22.   Evaluate the following Java expression:  9 - 15 \* 6 / 11 + 4
    - 53/11
  + 23.   Evaluate the following Java expression:  75 % 7
    - 5
  + 24.  Never forget that you should not compare two Strings with the == operator.  Suppose you have two String variables, x and y.  Give an expression that can be used to check whether or not the Strings x and y are identical.
    - x.equal(y)
  + 25.  What is "concatenation"?  What operator do you use to concatenate Strings?
    - Joing two strings together. Use +
  + 26.   What is the difference between System.out.print and System.out.println?
    - println moves the cursor to a new line have the output
  + 27.  Suppose you have a String variable called s.  What expression will return the number of characters in the String?
    - s.length()
  + 28. Give examples of "literals" of each of the following types:  String, char, long, int, float, double, boolean.
  + 29.  What "escape sequence" would you use in a String to indicate a "new line"?
    - \n
* Golub
  + 1.  What are the three logical operators?
    - == <, >
  + 2.  Write "truth-tables" for && and ||.
    - TFFF
    - TTTF
  + 3.   Is the following boolean expression true or false?    ((3 < 5) && !(1 > 14) && (-5 < -15)) || ((6 == 6) && !(2 == 2))
    - False
  + 4.  If s1 and s2 are variables representing Strings, what Java boolean expression is equivalent to "s1 is not the same as s2"?
    - s1 != s2
  + 5.  What statement must be included at the top of a file in order to use the Scanner in the file?
    - Java.util.Scanner
  + 6.  Write a java class called "UserInput".  In the main method, declare three variables:  an int, a float, and a String.  Name the variables "age", "weight", and "name".  Create a variable called "scan" of type Scanner, and set it equal to a new Scanner.  (Use the syntax shown in class).  Prompt the user to enter his/her age, weight, and name - read these entries in using the scanner and set the variables accordingly.  Then print the values of the three variables with appropriate labels.  For example:  "Name:  Frank     Age:  17    weight:  151.4".
    - import java.util.Scanner
    - public class UserInput {
      * public static void main(String[] args) {
        + int age, weight;
        + String name;
        + Scanner sc = new Scanner(System.in);
        + System.out.println(“What is your name?”)
        + name = sc.next()
        + System.out.println(“How old are you?”)
        + age = sc.nextInt()
        + System.out.println(“How much do you weigh?”)
        + weight = sc.nextInt();
        + System.out.println(“Name: “+name+” Age: “+age+” weight: “+weight);
      * }
    - }
  + 7.  Write a java class called "FahrenheitToCelsius".  The main method will ask the user to enter a temperature in Fahrenheit.  (Use a variable of type **double** to store the value.)  Then calculate the equivalent temperature in Celsius, and print out a message telling the user what you found.  [Recall: C = (5/9)(F-32).]  Hint:  Be careful about doing arithmetic with integers!  Check your program by entering 212 degrees.  The output should be 100.
    - import java.util.Scanner
    - public class FahrenheitToCelsius {
      * public static void main(String[] args) {
        + double f, c;
        + Scanner sc = new Scanner(System.in);
        + System.out.println(“Enter the temperature in Fahrenheit”);
        + f = sc.nextDouble();
        + c = 5 (float) / 9 \*(f - 32);
      * }
    - }
  + 8.  Modify the "FahrenheitToCelsius" question in the previous question so that the user can either go from F to C or vice versa.
  + 9.  FOR THIS EXERCISE, YOU SHOULD STRIVE TO AVOID REDUNDANT CODE!  Write a java class called "RequestInfo""  The main method will ask the user to enter his species.  If the user enters "dog", then ask him to enter the number of cats he has eaten this year.  If the user enters "cat", ask him to enter the number of hairballs he has coughed up this year.  If the user enters "predator", ask him to enter BOTH the number of cats he has eaten this year AND the number of hairballs he has coughed up this year.  If the user enters anything else (not dog, cat or predator), tell him that he is from another planet, and terminate the program.  If the user DID enter one of the three valid species (dog, cat, predator) then print out a report in the following format:
  + Species:  dog
  + Number of cats eaten:  54
  + Number of hairballs:  0
  + 10.  Write a program that computes the letter grade for a student based on his/her numerical total.  The program will read in the total and compute the letter grade based on the following:  to get an A the total must be at least 90.0.  To get a B it must be at least 80.0.  For a C, at least 70.0.  For a D, at least 60.0.  Less than 60.0 is an F.
  + 11a.  Write a program that asks the user to enter up to four scores from 1 to 10.  At the end the program will print out the total of the scores, but without including the highest score.  The catch is that at any time the user may enter 999 to indicate that he has no more scores to report.  For this version, you may not use any loops!   For example, here are a couple of possible runs of the program:
  + **Example 1:**
  + Enter score 1:  5
  + Enter score 2:  7
  + Enter score 3:  9
  + Enter score 4:  3
  + The total (without the highest) was: 15
  + **Example 2:**
  + Enter score 1:  8
  + Enter score 2:  9
  + Enter score 3:  999
  + The total (without the highest) was: 8
  + 11b. Do the same thing as in 11a but using a while loop.
  + 12.  Decide which of the following variable names are valid in Java:   dog, x11, \_tomato, big$deal, how&why, 22down, aBcDeFg, \_$\_\_$, under\_score, \_5\_$\_5\_hello13.
    - Dog – valid
    - X11 = not valid
    - \_tomato = valid
    - big$deal = not valid
    - how&why = not valid
    - 22down = not valid
    - aBcDeFg = not valid
    - \_$\_\_$ = not valid
    - under\_score = valid
    - \_5\_$\_5\_hello13 = not valid
  + 13.  Write a program that asks the user for an integer, call it n.  The program will then add up all of the integers from 1 to n and print out the total.  For example, if the user enters 4, then the output should be 10.  (Because 1 + 2 + 3 + 4 = 10).
  + 14.  The factorial of an integer is the product of all positive integers that are less than or equal to it.  For example, 4 factorial is 2 \* 3 \* 4 = 24.  Write a program that asks the user to enter a value, n, and then prints n factorial.
  + 15.  Write a program that asks the user to enter two values: x and y.  You must then compute the product of all integers from x to y.  For example, if the user has entered 10 and 7, then the output should be 5040 (because 7 \* 8 \* 9 \* 10 = 5040).
  + 16.  Write a program that "simulates" the reading password process you go through while logging into a computer account.  The program will ask for a password, compare the value against two possible passwords (that are built-in), and print "Welcome" if the password provided by the user is valid. Otherwise, ask the user to enter the password again.  (This process repeats.)
  + 17.  Write a program that asks the user to enter the number of rows and columns.  It will then print out a rectangular grid of asterisks.  For example, if the user has entered 6 rows and 3 columns, then the output should be:
  + \*\*\*
  + \*\*\*
  + \*\*\*
  + \*\*\*
  + \*\*\*
  + \*\*\*
  + 18. CHALLENGE QUESTION  Write a program that asks the user for a size (an integer).  The program will then print out four different triangles made out of asterisks of that size.  (You will need to print out spaces sometimes in front of the asterisks.)   This is a "challenge" problem. Below is the output if the user selected size 4:
  + \*\*\*\*
  + \*\*\*
  + \*\*
  + \*
  + \*
  + \*\*
  + \*\*\*
  + \*\*\*\*
  + \*\*\*\*
  + \*\*\*
  + \*\*
  + \*
  + \*
  + \*\*
  + \*\*\*
  + \*\*\*\*
  + 19.  Write a program that asks the user for a size, and then prints out a multiplication table of that size.  (You don't have to worry about spacing correctly, just try to get the numbers to all come out on the right rows.)  For example, if the user requests size 4, the output should be:
  + 1 2 3 4
  + 2 4 6 8
  + 3 6 9 15
  + 4 8 12 16
* 1.  Go back to the loop questions from the previous week and implement them all using for-loops instead of while loops!
* 2.  What will the output be: **(x = 2);**
* int x = 2, y = 7;
* if (y < 1 && x++ < 7)
* System.out.println("hello");
* System.out.println("x = " + x);
* 3.  What will the output be: **(x = 7, y = 10, z = 56)**
* int x = 7;
* int y = x++ + 3;
* int z = 7 \* --x + 4;
* System.out.println("x = " + x + ", y = " + y + ", z = " + z);
* 4.  What will the output be: **(7)**
* int x = 5;
* x += 7;
* x -= 2;
* x /= 2;
* x \*= 8;
* x %= 11;
* System.out.println(x);
* 5.  Write a chart showing the precedence of all of the following operators:   = , < , ==, &&, ||, !=, ++, + (addition), +=, --, \*
* 6.  If two operators occur in the same expression, and they are on the same level in the precedence chart (it is a "tie"), how do you decide which operator gets evaluated first? **Left to right**
* 7.  Using parentheses indicate the order in which each of the following expressions will be evaluated or state that the expression represents an invalid expression.  You may assume that all variables are of type int.
* a. ((x / y )\* z) % w
* b.  (x ++) + (y ++)
* c.  x + y + z – ((w % p) \* 2))
* d.  (x < y) || ((z > m) && (y <= 4 ))
* 8.  Write psuedocode for a program that computes the number of digits in an integer.  For example, if the user enters 1792, the output will be the value 4.
* 9.  Write psuedocode for a program that reads a sequence of integer values and decides whether or not it is a decreasing sequence.  The program will first read in the number of values to process, followed by the values themselves.  The output will be "Yes" if the sequence is decreasing, and "No" otherwise.
* 10.  Which of the following code fragments are OK, and which will cause problems?
* a.  int x = 52;
* double y = x; **ok**
* b.  double x = 14;
* int y = x; **no**
* c.  int x = 7;
* long y = x; **ok**
* d.  long x = 17L;
* short y = x; **no**
* 11.  Show how to use "explicit casting" to force the troublesome examples in the previous question to work. (NOTE: This is something that is in the posted notes.)
  + int y = (int) x;
  + short y = (short) x;
* Evan Golub
  1. What is meant by "the state" of an object?
     1. **The data**
  2. What is meant by "the behavior" of an object?
     1. **The method**
  3. How is the state of an object represented in the corresponding class definition?
     1. **Instances variables**
  4. How is the behavior of an object represented in the corresponding class definition?
     1. **Instances methods**
  5. What is a "reference" variable? the object.
     1. **A variable that refers to the memory address of an object**
  6. Given the code fragment **String q = new String("Cat");** and the code fragment **double x = 7.9;** is it correct to say that "q is a String object"? Is it correct to say that "q refers to a String object"? Is it correct to say that "x is a double"? Is it correct to say that "x refers to a double"?
     1. **NYYN**
  7. Consider the assignment statement: x = y;   
     If the variables are of type "int", does the integer get copied? If the variables are of type "String", does the String get copied?
     1. **Y N**
  8. True/False: In order to run a static method, you must run it for a particular object (the current object).
     1. **False**
  9. True/False: In order to run an instance method, you must run it for a particular object (the current object).
     1. **True**
  10. True/False: An instance variable can be declared as "static".
      1. **False**
  11. True/False: The same class can contain some members that are static and some members that are non-static.
      1. **True**

Evan Golub

* 1. What is "the heap"?
     1. **The permanent memory**
  2. Draw the memory map (including the "stack" and the "heap") for the following code fragment:
     1. double x = 7.9;
     2. double y = x;
     3. String q = new String("Cat");
     4. String r = q;
     5. String s = new String(r);
  3. What is actually stored in the variable "q" in the previous question?
     1. Memory address
  4. Given the code fragment above, is it correct to say that "q is a String object"? Is it correct to say that "q refers to a String object"? Is it correct to say that "x is a double"? Is it correct to say that "x refers to a double"?
     1. NYYN
  5. How many String objects are actually created (instantiated) in the code fragment above?
     1. Two
  6. After the code fragment above has been executed, decide which of the following expressions are true and which are false:
     1. x == y
        1. True
     2. q == r
        1. True
     3. r == s
        1. false
     4. q == s
        1. false
     5. q.equals(r)
        1. True
     6. r.equals(s)
        1. True
     7. q.equals(s)
        1. True
  7. What is aliasing?
     1. Copying the memory address to certain data
  8. What is meant by the term "garbage" in Java? Write a code fragment that creates garbage.
     1. Data that is not being pointed to
     2. String g = new String(“Hey”);
     3. g = Yo”;
  9. Write a class called "Cat". Invent a few "instance variables" that make sense for a Cat. Include a constructor that allows the user to specify values for all of the instance variables at the moment the Cat object is instantiated (created). Include a toString method that produces a String representation of the state of the Cat. Include an equals method that makes sense to you. Write some other methods (behaviors) that Cats should be able to do. Make sure that you have practiced some method(s) that require parameters to be passed in. Make sure that you have practiced some methods that return a value.
     + Cat() {
       - int livesLeft;
       - String color;
       - String name;
       - public Cat(int lives, String col; Sting name) {
         * livesLeft = lives
         * color = col;
         * name = name;
       - }
       - public String toString() {
         * return “Name: ”+name+” Color: ”+color+” Lives: ”+livesLeft;
       - }
     + }
  + Write a driver for the Cat class that you created above. The driver should create a few Cats and then test out all of the methods that you have written. Try to test everything thoroughly!
  + Write a class whose members include a few static variables and a few static methods. Then write a driver that calls the static methods and uses the static variables in some way.
* Evan Golub
  1. Which kinds of variables are given default values if you do not initialize them?  (Local/ instance/static?  Which ones?)
     + Instance and static
  2. What default values are used for the variables mentioned above?
     + Null; 0
  3. When writing JUnit tests, is it better to write many small tests, or a few long ones?
     + Many small tests
  4. What is JUnit?
     + A debugging library
  5. Give an example of the correct syntax for **assertEquals** and **assertTrue** in a JUnit test.
  6. If an individual JUnit test has several separate assertions in it, will the test continue after one of the assertions fails?
     + No
  7. If a JUnit test suite has several different JUnit tests in it (separate methods), will the rest of the tests run after one of them fails?
     + Yes
* Evan Golub
* 1.          In what kind of methods does it make sense to use "this"?
  + nonstatic method
* 2.          What does "this" refer to?
  + The object that the method is acting on
* 3.          If you do not write any constructors, what values will instance variables of the following primitive types be assigned:  int, double, boolean, char   ?
  + int = 0
  + double = 0.0
  + Boolean = false;
  + Char = “”;
* 4.          If you do not write any constructors, what values will instance variables that are references by assigned?
  + Null;
* 5.          Under what circumstances will Java provide a default constructor for you automatically?
  + If you do not have any constructor methods and you create a new object
* 6.          What is a copy constructor?   Give an example.
  + A constructor that is used to copy another object into a new object
* 7.          What is a Stack (in general, not just in Java)?
  + Temporary Memory used to store addresses and
* 8.          When you push an entry into the stack does it go on the top or bottom?
* 9.          When you pop an entry from the stack, does it come off the top or bottom?
* 11.          True/False - in java, when you pass a reference variable as an argument to a method, it is possible for the method to modify the object to which the variable refers.
* 12.          What does API stand for?
  + Application Program Interface
* 13.          If someone showed you a Java class, how can you quickly identify which members were part of the API for that class?
* 14.          If a member is declared as "public", can it be accessed from inside the same class?
  + yes
* 15.          If a member is declared as "public", can it be accessed from another class?
  + yes
* 16.          If a member is declared as "private", can it be accessed from inside the same class?
  + yes
* 17.          If a member is declared as "private", can it be accessed from another class?
  + no
* 18.          What is a "setter"?
  + A method that assigns some value to an object
* 19.          What is a "getter"?
  + A method that retrieves some value from an object
* 20.          Explain why it is important to limit the number of "public" members.
  + You’re code becomes vulnerable
* 21.          Name and describe the two visibility specifiers that you should know at this point.
  + Public/Private
* 22.          True/False - if you change a class in such a way that the API changes, then other classes which depend on this one will have to be re-coded.
  + False
* 23.          True/False - if you change a class without modifying the API, then other classes which depend on this one will have to be re-coded.
  + True
* 24.   What package is the Scanner class located in?  What is the fully qualified name of the Scanner class?
  + Java.util.Scanner
* 25.          What is accomplished when you type "import java.awt.Color;" at the top of a file?
  + You can access the java.awt.Color library
* 26.          What is accomplished when you type "import java.awt.\*;" at the top of a file?
  + You can access all the libraries held in the java.awt package
* 27.          Which java package is automatically imported in it's entirety into every Java program you write?
  + Java.lang.\*
* 28.          What method of the String class can be used to pick out one particular character in the string?
  + charAt()
* 29.          What method of the String class can tell you how many characters are in the String?
  + length()
* 30.          What method of the String class can be used to compare to Strings for alphabetical order?
  + compareTo()
* 31.          What method of the String class can select a portion of an existing String?
  + substring()
* 32.          Write a method called "count".  The method should be public and static.  It takes one parameter, (a reference to a String).  The method will return an *int*.  The return value should be equal to the number of Xs that appear in the String.  For example, if the parameter is:  "XaXXXbXXc" then the return value would be 6
  + public static int count(String str) {
    - int totX = 0;
    - for (int x = 0; x < str.length; x++) {
      * if (str.charAt(x) == “X”) {
        + tot++;
      * }
    - }
    - return totX;
  + }
* Evan Golub
  + 1. Write some faulty code that generates a null-pointer exception, catch the exception immediately and print out "exception caught" in your catch block.
  + 2. Write some code that prompts the user for a numerical value, and reads their input into an int variable. Run the program, and try entering some text (like "cat") instead of a number. Notice what kind of exception is thrown. Now modify your program so that it catches this exception, and instead of crashing the program, have it tell the user that he/she must enter a NUMBER, and then prompt them for input again.
  + 3. Write a method called smallSum that takes two int parameters, x and y. If the absolute value of the sum of the integers is more than 100, throw an ArithmeticException, passing the String "I don't like big numbers" to the constructor of the exception. If the sum is less than 100, then return the sum. Write a quick driver to test out your method. After making sure everything works correctly, modify the driver so that it catches the exception and prints out the message that was passed to the exception's constructor, but doesn't crash the program.
  + 4. Explain the relationship between exception handling and the call stack. What happens if an exception is thrown but not caught anywhere in your program?
  + 5. Under what circumstances will the finally block run
* Evan Golub
* 1.  How many ints are created by the statement:  int[] a = new int[5];
* 2.          How many Strings are created by the statement:  String[] a = new String[5];
* (Hint:  The answer to this question and the previous question are different!)
* 3.          Are the elements of an array of primitives automatically initialized?  If so, to what values?
* 4.          Are the elements of an array of references to objects initialized?  If so, to what values?
* 5.          Draw the memory diagram for each of the following code fragments:
* a.     int[] a = new int[4];
* b.     String[] b = new String[4];
* for (int i = 0; i < b.length; i++)
* b[i] = "value: " + i;
* 6.          Write a class that has an instance variable which is an array of Cat objects, called kitties.  Write a method that returns a reference copy of kitties.  Write a method that returns a shallow copy of kitties.  Write a method that returns a deep copy of kitties.
* 7.          Suppose you are passing (or returning) an array of primitives to/from a method.  Is it safe to make a reference copy only?
* 8.          Suppose you are passing (or returning) an array of references to immutable objects to/from a method.  Is it safe to make a reference copy only?  Is it safe to make a shallow copy?
* 9.          Suppose you are passing or returning an array of references to mutable objects to/from a method.  Is it safe to make a reference copy only?  Is it safe to make a shallow copy?
* 1.
* a.  Write a code fragment that creates a two-dimensional ragged array of ints with 3 rows,  initialized with the following data:
* 5        8     9
* 4    2   13  15   17
* 0     1
* b.  After you have created this array, write code that will print the contents in the same format that you see above.
* c.  Draw the memory map for this array.
* 2.
* a.        Write a method called catDuplicator, with the following prototype:
* public Cat[][] catDuplicator(int[] rowSizes, Cat c)
* The method will create a two-dimensional ragged array, using the array"rowSizes" to determine how many rows there are, and how long each row must be.  Each element of the array that gets created will refer to the very same cat, c.
* For example, if the array rowSizes contains the data:   5    7    2    1
* then the return value would be a two-dimensional ragged array with four rows.  The first row would be size 5, the second row would be size 7, etc.  Each element of the two-dimensional array must be a reference to c.
* b.  Draw the memory map for the method above.
* 3.           Write code that asks the user for a value (n), and then creates an n by n two-dimensional array of ints.  Fill the array with a multiplication table.  For example, if n is 3, the table should be:
* 1     2     3
* 2     4     6
* 3     6     9
* 4.          a.
* Write a method which has the following prototype:
* public static int[] linearize(int[][] array)
* The method linearizes the two-dimensional array by returning a one-dimensional array with all the elements of the parameter (selected row-by-row).  The original array cannot be modified.
* b.  Write a JUnit test that tests your method.
* 5.          Write a method called "deepCopy" that takes a parameter (a two-dimensional array of references to StringBuffer objects) and returns a deep copy.  (You must make copies of the StringBuffer objects themselves since they are mutable.)
* Evan Golub

What is a Java interface?

2. What is polymorphism?

3. Suppose you have an interface called CanDance, and three classes (Student, Penguin, and Cow) all of which implement the CanDance interface.  Also assume that there is a method available with the following prototype:

public static doSquareDance(CanDance a)

Decide which of the following code fragments are reasonable:

a.  CanDance x = new CanDance();

b.  CanDance y = new Student();

c.  Student z = new CanDance();

d.  Student z = new Penguin();

e.  CanDance a;

     a = new Student();

     a = new Penguin();

     a = new Cow();

f.  Penguin b = new Penguin();

     doSquareDance(b);

g.  Student c = new Student();

     doSquareDance(c);

h.  Cow d = new Cow();

     doSquareDance(d);

i.  CanDance e = new Student();

     doSquareDance(e);

4.

a.        Write a Car class.  (Use your imagination for fields.)

b.      Now write an interface called "CanFixCars" with two method prototypes:

public void fixFlat(Car c);

public void fixRadiator(Car c);

c.  Write three classes:  CSMajor, MathMajor, and CEMajor, each of which implements the "CanFixCars" interface. Be creative when implementing the methods.  How do you think a Math Major would fix a flat tire?

d.  In a separate class, write a static method with the following prototype:

public static fixCar(Car c, CanFixCars repairPerson)

The method should somehow determine what is wrong with the car (is it a flat tire, a broken radiator, or something else) and have the repair person fix the car by calling the repair person's fixFlat or fixRadiator methods.

e.        Finally, write a main method that will create several broken cars, create several students of various kinds, and have the students fix the cars.  (i.e.:  make several calls to your fixCar method.)

* + 1. What does "method overloading" mean?
    2. For each pair of methods, decide whether or not both methods could be implemented in the same class.  If so, could there ever be a situation where an ambiguity could arise?

1. void f(int x)
2. void f(int y)
   * + 1. int f(int x)
       2. void f(int x)
3. void f(int x)
4. void f(double x)
5. void f(int x, int y)
6. void f(int y, int x)
7. void f(int x, String y)
8. void f(String y, int x)
9. void f(int x, double y)
10. void f(double x, int y)

3.          Using a switch statement, write a method that accepts one parameter, a char, which is expected to be a capital letter from A to Z.  The method will return the number of "pen-strokes" that are required to draw the letter.  For example, the letter Z requires three strokes (top, bottom, diagonal section) and the letter P requires two (the straight part, and the curved part).  You must avoid duplicative code.  If the parameter is not a letter in the range A through Z, then have the method return 0.

4.    If you write a switch statement like this:

* + switch(x) {...}
  + What types could the variable x be?
* 1.             What are the three components of the M-V-C pattern?
* 2.             What are the roles of each of the three components of the M-V-C pattern?
* 3.          When you use a public class inside of it's own package do you need to use a fully qualified name?
* 4.          Can you use a public class outside of it's package?  Do you need to use a fully qualified name?
* 5.          When you use a non-public class (with no visibility specified) inside of it's own package, do you need to use a fully qualified name?
* 6.          Can you use a non-public class (with no visibility specified) outside of it's package?
* 7.          If you write a class without specifying a particular package, which package will it become part of?
* 8.          What syntax would you use at the top of a source-code file to specify that the class belongs in a subpackage called "stuff" that is part of a larger package called "junk"?
* 9.
* a.        Just for practice, write your own MyStack class which stores Integer objects.  Use a private array of Integer objects to store the data.  You should include the methods "push" and "pop".  Write a JUnit test to check if everything is working.
* b.      Now modify your code so that it uses an ArrayList<Integer> instead of an array.
* c.       Now that you've built those JUnit tests, modify your JUnit test so that it works with objects of type Stack<Integer> instead of MyStack.
* 10.          Write code that stores 50 randomly generated floating point numbers in an ArrayList<Double>.   Use a regular for-loop to cycle through the values and remove any that are less than 0.5.  Then use a for-each loop to print out the values that remain.
* 11.          Write code that creates an ArrrayList and then uses a for-each to cycle through.  Try inserting something into the arrayList in the loop where you are using the Iterator.  What happens?
* 12.          What do you accomplish by using "extends"?  (What does it mean to write:  public class A extends B)?
* 13.          What does it mean to say a class is derived from another class?
* 14.          What does it mean to say a class "inherits" members from another class?
* 1.          What is method overriding?   How is it different from overloading?
* 2.       Draw an inheritance diagram (with the arrows going the correct direction) illustrating how the 'Animal Kingdom' is organized. (It doesn't have to be very scientific, unless you are an overachiever.)
* 3.   Write a class called 'Animal'.  Put in a couple of fields and methods that would make sense for any Animal.  Now write a couple of classes that extend Animal (Cat, Dog, Squid, whatever...)  Have these extensions have a few fields and/or methods of their own.  Also practice having some of your derived classes override some of the Animal methods.  Write a good set of JUnittests that thoroughly test your project.
* 4.  Write a correct equals method for your Animal class.  Be sure your parameter is type Object (so that you are over-riding the equals method from the Object class).  Be sure to do the null check, and the class check.  Then don't forget that you
* will need to use a casting operator to access the instance variables for the parameter!
* 5.          How is "super" used in a constructor of a subclass?  What does it do?
* 6.          When do you need to use "super" to access a member of the base class?
* 7.    Assume that you have a class called "Politician" with two fields:
* int numberOfLiesTold
* int numberOfBabiesKissed
* Now assume that you have a class called "Democrat" that extends Politician and has two fields of it's own (I'll let you choose fields that you feel fit.)
* Assume that the default constructors for Politician and Democrat will initialize the fields with some reasonable values.  Now draw a memory diagram for the following code fragment:
* Politician x = new Politician();
* Democrat y = new Democrat();
* Politician z = y;
* 8.          In the previous question, which part of a Democrat object is actually initilized first?
* 9.          In the previous example, can you use a Politician anywhere a Democrat is expected?  Can you use a Democrat anywhere a Politician is expected?
* 10.  Using the classes from the previous question(s), assume you have executed the following three lines:
* Politician p = new Politician();
* Democrat d = new Democrat();
* Politician x = new Democrat();
* Which of the following will compile?  Which will throw exceptions when you run them?
* Politician a = p;
* Democrat b = p;
* Politician c = d;
* Democrat e = p;
* Politician f = (Politician)d;
* Democrat g = (Democrat)p;
* Democrat h = x;
* Democrat h = (Democrat)x;