

CSCI 131 Midterm Exam 2 – Review Questions

This sheet is intended to help you prepare for the second exam in this course. The following topics have been covered so far in the course. The exam will focus on chapters 1.4, 1.5, 2.1, and 2.2, of the textbook, up to the lectures in week eleven and all of the labs and projects that have been assigned so far. The exam assumes you have mastered all material covered in the previous exam, especially loops and selection statements. Output formatting (printf) will not be emphasized, but you should know how to use basic output formatting with printf. The following is a partial list of topics that are likely to be emphasized on the exam:

1. Libraries and Clients

2. 2D Arrays

- Declaring 2D arrays, accessing array elements.
- Iterating over rows and/or columns of a 2D array

3. Input and Output

- Using input file streams (In), and output file streams (Out).
- File input/output: opening and closing files
- Interactive input/output: prompting for then reading input

4. Functions/Static Methods

- Writing and invoking functions.
- Function signatures/prototypes.
- Using arrays as parameters.
- Call by value parameters and parameters that are in effect reference (arrays) parameters.

5. Scope

- Precedence rules when variables have the same names.
- Scope rules for functions and other blocks of code.

Practice Problems

1) A function, `someFunc`, has two formal parameters named `alpha` and `beta`, both of type `int`. The function returns a boolean value. Write an appropriate signature for the function.

2) Consider the function definition:

```
public static int demo (int intVal, double floatVal) {
    intVal = intVal *2;
    floatVal = (double)intVal + 3.5;
    return intVal;
}
```

Suppose the caller has variables `myInt` and `myFloat` whose values are 20 and 4.8 respectively. What are the values of `myInt` and `myFloat` after return from the following function call?

```
myInt = demo(myInt, myFloat);
```

3) Given the function definition:

```
public static int mystery( double someVal ) {
    if (someVal > 2.0) {
        return 3 * (int)someVal;
    } else {
        return 0;
    }
}
```

What is the value of the expression `mystery (4.2) ?`

4) a) Write a function that returns 10.0 raised to the integer power specified by parameter `n`. (Recall that 10.0 raised to power 0 is 1.0, 10.0 raised to power 1 is 10.0, 10.0 raised to power 2 is 100.0, and so on.) The function should match the following signature. You may not use any Math library functions – you should instead compute the answer using a loop.

```
public static double tenToThePower( int n )
```

b) Write a single JAVA statement that invokes the above function to compute 10.0 to the power 8 and store the answer in the variable `b`.

5) Given the following array and variable declarations:

```
int[] ssn = new int[100];  
int numPatients;
```

Write a loop that reads from the keyboard the social security number for each of the patients and stores the values in the appropriate position in the appropriate array. Stop reading numbers once the user enters 0 or a negative number. You can assume the user will enter at least one valid social security number and no more than 100 of them.

6) Given the declaration

```
int[][] checkers = new int[8][8];
```

a) Write a statement that will store the number five in the 2nd row, 3rd column of the array.

b) Write JAVA code to store the numbers 1-8 in the first row, 9-16 in the second row, 17-24 in the third row, and so on.

7) Write a function called `arrayRev` that reverses an array of integers. The function should take two parameters, `source` and `destination`, both of which have an appropriate type for an array of integers. The function does not return anything. The function should put the last integer of `source` into the first position in `destination`, the next-to-last integer of `source` into the second position in `destination`, and so on. You can assume both arrays are the same size.

8) Write a Java (static method) function named `copyUniqueStrings()` that copies a list of strings from one array to another array, but ensures that there are no duplicates in the resulting array. The function should have three parameters: an array `src` of `Strings`, the number `n` of valid elements in that array, and an array `dst` of `Strings`. The function should examine each of the first `n` strings in `src`, putting each string into `dst`, but only if that `String` does not already appear in `dst`. The function should return a count of how many unique strings were copied to `dst`. You can assume the length of `src` is at least `n` and that `dst` is large enough.

Use `FindWithinArray (#15)` to make this easier.

9) Write a function that checks if a given two arrays of integers have identical contents. The function should have the following signature:

```
public static boolean identicalLists(int[] one, int[] two)
```

The function should return true only if the two lists have the same contents.

10) Below are snippets of code that are trying to replace each of the leading 1 numbers in an integer array with a 3. For each loop, say whether it is correct and explain why.

<pre>int[] name = {1,1,1,1,2,2,2}; int ct = 0; while (name[ct] == 1) { name[ct] = 3; ct++; }</pre>	<pre>int[] name = {1,1,1,1,2,2,2}; int ct = 0; while (name[ct] == 1) { ct++; name[ct] = 3; }</pre>
--	--

11) Consider the following program:

```
public class Circle { // line 1
    // line 2
    public static void main (String[] args) { // line 3
        double radius = 52.0; // line 4
        // line 5
        StdOut.println ("The perimeter of the circle " + // line 6
            "with radius " + radius + " is " + // line 7
            2 * Math.PI * radius); // line 8
        // line 9
        StdOut.println ("The perimeter of the circle " + // line 10
            "with radius " + 31.0 + " is " + // line 11
            2 * Math.PI * 31.0); // line 12
    } // line 13
} // line 14
```

Use the above program to answer the questions in parts a and b.

a) Notice that the lines 6-8, and 10-12 have a similar pattern of code. Write a function, `fix()` that captures this similarity, so that each of these sets of 3 lines can be replaced with a single call to `fix()`. Your function should have only **one** parameter and should **not** return a value.

b) Rewrite lines 6-12 of the `main()` function using two distinct calls to your function, `fix()` to replace lines 6-8, and 10-12.

12) Write a program that inputs a series of month names and temperatures from a file, and outputs a bar chart (using stars) of the temperatures. The month name should be printed to the left of the corresponding bar. You can assume all of the month names have 3 characters and all of the temperatures are in the range 0 to 120. Because it is hard to display 120+ characters wide on the screen, you should have each star (the character ' * ') represent 3 degrees. That way, the bars will be at most 40 characters wide plus the labels. An example is shown below. Note how the temperatures are rounded to the nearest appropriate number of stars. The input file is named "temps.dat". Use a loop to process the input file. Write a function to help you if you like.

For example, if the input file contained:

```
Jan 5
Mar 20
May 105
Dec 35
```

Then the output of the program should look like this:

```
Temperatures (each star represents 3 degrees)
Jan **
Mar *****
May *****
Dec *****
```

13) Assume two arrays of the same length have already been declared and initialized. The first array, names, contains student names. The second array, ages, contains the corresponding age of each of those students.

(a) Write code to print out the name and age of every student, one per line, using the following format:

```
Alice is 5 years old.
Bob is 3 years old.
Carol is 7 years old.
Grace is 5 years old.
```

(b) Write code to instead print out only the names of the students younger than 6 years old, using the following format: **Young students include: Alice Bob Grace.**

(c) Write a function to count the number of students who are at least x years old and younger than y years old. Your function should be named countAgeRange and it should have appropriate parameters and return type.

(d) Write a function named ageOfStudent that will find the age of a particular student, given a name. It should take three parameters: an array of names, an array of ages, and the name of the student being sought. For example, if invoked like this:

```
int x = ageofStudent(names, ages, "Alice");
```

then the function should return Alice's age based on the information in the names and ages arrays. If the name "Alice" does not appear in the array, the function should return -1 instead to indicate that the name could not be found.

- 14) A file named "temperature.txt" contains temperature data, but unfortunately is a mix of both Fahrenheit and Celsius temperature systems. For example, "27.5 C 98.7 F 350 F 35 C ...". Write a snippet of code that reads all of the data from the file and stores it into parallel arrays (one for the numbers, one for the "F" and "C" symbols). Then, convert all of the data to either Fahrenheit or Celsius, whichever was more common (if they were equally common, pick either system). The formula to convert F to C is: $C = (F - 32) * 5/9$. The formula to convert the other direction is $F = 32 + C * 9/5$. When finished, print all the data into a new file, "normalized-temperature.txt". The data in this new file should be the same order as it was in the original file, but all in one temperature system. Print one temperature per line, preferably with one decimal place (though that's not important for this exam).
- 15) Write a function named "findWithinArray" that takes as parameters an array of strings, and a target string. The function should search the array to see if the target string is contained within it. If so, the function should return the array position where the target was found. If the target was not found anywhere in the array, the function should return a -1 to indicate "not found".
- 16) In an election with 5 candidates, the names of the candidates are stored in a file named "candidates.txt". Read these into an array. You can assume each candidate name is one word, like "Alice" or "Sekou". A second file, named "votes.txt", contains voting data. Each line looks like "Alice 15" or "Sekou 7", which counts as 15 votes for Alice or 7 votes for Sekou. Read all of this data, and tally up the votes for the 5 candidates. Print the totals, and print the name of the winner. Any votes for someone not on the candidate list should be ignored, but counted as "spoiled votes". Print a count of how many spoiled votes there were. Hint: Use two parallel arrays, and make use of the "findWithinArray" function from the previous exercise.
- 17) **[this question is taken directly from a previous year exam]** Suppose a file contains a list of names (e.g. students in a class) and several integers for each name (e.g. grades for that student). An example of such a file is shown below: each line starts with a student name, followed by grades for that student. Write a function that reads all of the names and numbers from such a file into two arrays. The function should be named **readData()** and it should have three parameters: the name of the file, an array to hold all of the names, and a two-dimensional array of integers to hold the numbers. You can assume the arrays will be the appropriate sizes for the information in the file. The function should return nothing. Note: for a 2D array, `array.length` gives the number of rows, and `array[0].length` gives the number of columns in row 0.

Example input file, showing format of the lines. The spacing is not important.

Alonzo	9	8	11	8
Betty	9	10	8	11
Leslie	12	9	10	8
Grace	10	8	12	9
Niklaus	9	7	8	10
Frances	11	13	8	5

18) [this question is taken directly from a previous year exam] Often we have an array filled with Strings (e.g. a list of words) and we would like to know if a particular word is in the list. Write a function named **doesContain()** to do just that. Your function should have three parameters: an array, names, containing Strings, the number, n, of elements in the array to be searched, and a target String, s. If the word s is one of the first n items of list names, your function should return true. Otherwise your function should return false. You can assume the array's length is at least n.

19) [this question is taken directly from a previous year exam] Write a program to simulate a simple dice game. In each round, a pair of dice is thrown. If the sum of the dice matches a special number, the round is counted as a win. Your program will simulate this game, printing each roll and counting the number of winning throws. The number of rounds and the special number are given to the program as command line input.

```
[radius]$ java Dice 6 9
Rolling ...5 6
Rolling ...5 4
Rolling ...3 3
Rolling ...4 2
Rolling ...3 2
Rolling ...3 6

After rolling pairs of dice 6 times,
the sum 9 was matched 2 times.
```

20) [this question is taken directly from a previous year exam] Write code to read a sequence of integers from a file named full.txt and print them into a file named compress.txt, except your code should remove consecutive repeated values in the input. For example, if full.txt contains

1 2 2 1 5 1 1 7 7 7 1 1 1 1 1 1 1 1

then your program should store into the file compress.txt these values:

1 2 1 5 1 7 1

You can assume the file contains only positive integers.

21) [this question is taken directly from a previous year exam] A traveler wishes to visit various cities, at random, by train. There are N cities in all. The cities are numbered (0, 1, 2, etc.), and each city has a name (Shanghai, Beijing, Shenzhen, and so on...). From each city, there are three possible destination cities. From their home city, the traveler will pick a random first destination reachable from home, then pick a random second destination reachable from the first destination city, and so on. If the traveler ever arrives back at their home city, or ever arrives at a city they have already visited, then the journey ends immediately. Otherwise, if the traveler visits each of the N cities exactly once, then the journey ends and it is called a “*grand tour*”.

Example railway.txt file containing 7 city names and three destinations for each:

Shanghai	2	4	5
Beijing	2	0	4
Shenzhen	3	5	6
Nanning	0	1	5
Chengdu	3	4	6
Zhengzhou	0	6	1
Changsha	0	5	4

Write a program to generate a random list of cities for our traveler to visit. The number N is a command-line parameter. A file named **railway.txt** (see example at right) contains the city names and the three destinations reachable from each city. Your program should start by reading the information from this file and storing it into appropriate variables, arrays, and so on. The program should ask the user to enter their home city, then randomly generate and print out a list of cities to be visited. Finally, the program should say whether or not the journey is a grand tour. Hint: You may use any of the functions described in other questions on this exam. The **readData()** function and **doesContain()** from the previous two questions should be very useful. You can assume those functions exist and work, even if you didn't solve those question.

- 22) [this question is taken directly from a previous year exam] Write a program to simulate a simplified version of the game Tokaido, in which two friends Aki and Baku walk N steps along a road to their destination. The number N should be a command-line argument. To play, each player rolls their own standard 6-sided dice. Whichever player rolled the **lower number** takes that many steps forward along the road. If the player moving would land on the same spot as the other player, they should take one extra step forward to land on their own space (see last step of example on the right). If the players roll the same number, neither player moves (see example on left). Next, the players roll again, and repeat the process until one of the players has taken at least N steps and becomes the winner. The program output should look like the examples below.

```

Example output:
java Tokaido 8
Aki rolls 6 and Baku rolls 5.
  Baku walks 5 steps.
  Aki has walked 0 and Baku 5.
Aki rolls 4 and Baku rolls 4.
  Aki has walked 0 and Baku 5.
Aki rolls 3 and Baku rolls 6.
  Aki walks 3 steps.
  Aki has walked 3 and Baku 5.
Aki rolls 5 and Baku rolls 4.
  Baku walks 4 steps.
  Aki has walked 3 and Baku 9.
Baku wins!

```

```

Example output:
java Tokaido 7
Aki rolls 3 and Baku rolls 5.
  Aki walks 3 steps.
  Aki has walked 3 and Baku 0.
Aki rolls 4 and Baku rolls 2.
  Baku walks 2 steps.
  Aki has walked 3 Baku 2.
Aki rolls 6 and Baku rolls 4.
  Baku walks 4 steps.
  Aki has walked 3 and Baku 6.
Aki rolls 3 and Baku rolls 5.
  Aki walks 4 steps.
  Aki has walked 7 and Baku 6.
Aki wins!

```

Hint: The expression `StdRandom.uniform(0, x)` will pick a random integer between 0 and x, including 0 but not including x.

- 23) [this question is taken directly from a previous year exam] Given an M-by-N grid of integer values, write a code fragment that finds matching row neighbors and *reduces* them. A reduction consists of changing the left cell of the pair to be the sum and the right cell to be zero. (However, a pair of zeroes is not reducible.) Continue processing the grid until no more reductions can be made. The grid has been declared as an array:

```
int [][] grid = new int [M][N];
```

You may also assume the grid has been appropriately filled with values. Note: If you can't solve the full problem, try to solve a simplified version of the problem. For example, just visit each cell in the grid once and perform as many cell reductions as possible.

