CSCI 131, Midterm Exam 1 – Review Questions

This sheet is intended to help you prepare for the first exam in this course. The following topics have been covered in the first 5 weeks of the course. The exam will cover most of chapters 1.1 – 1.4 of the textbook, and all of the lectures up through and including the Thursday of week 5. It will also cover labs 1 - 5 and projects 0, 1 and 2. Each of the following topics may appear on the exam.

1. Basic syntax & style
    - Overall structure of a Java program
    - Data types: float, double, int, char, String, arrays
    - Variables and constants (final static)
    - Assignment statements
    - Arithmetic Expressions (operators for each of the data types, precedence rules, associativity)
    - Type conversion and type casts
    - Using Java libraries

2. Input and Output
    - Using command line arguments.
    - Sending output to the console (StdOut and System.out).
    - Interactive input/output: prompting for then reading input (simple StdIn).

4. Conditional Statements
    - Basic syntax: if, if…else…, if...else if…else if … etc.
    - Nested conditionals
    - Use of curly braces and proper indenting
    - Boolean (true/false) expressions and variables
    - Relational operators: >, <, >=, <=, ==, !=
    - Logical operators: &&, ||, !

5. Loops
    - while loops, for loops, do-while loops
    - controlling loops: count, event, sentinel, etc.
    - nested loops

6. Arrays
    - Declaring arrays, accessing array elements.
    - Initializing arrays.
    - Iterating over arrays.
    - Arrays with meaningful index values.
    - Parallel arrays.

Practice Problems

The following problems are intended to help you study for the exam, however topics not covered here may be on the exam as well. Use your text, class notes, labs and assignments to review as well. **The textbook site has many good review problems and contains numerous exercises with answers.**

1) Which of these is a valid ***assignment*** statement (circle all correct answers):

```
players = teamA + teamB;

int  players = team  A + team  B;

int players = (int) teamA + (int) teamB;

int = teamA + teamB;
```

2) Assume the following prototype (heading/signature) is in the Math library:

```
        double Cube(double number);
```

a) What does the word "`double`" at the beginning of the line refer to?

b) Write a single line of code to declare a double variable named ***`answer`*** and assign to it the cube of `7.3`. Use the above library function.

c) Write a statement to find the cube of the average of x and y and print the result. Use the above library function. Assume x and y are integer variables that are already declared and initialized.

3) Compute the value of the following expressions:

a) `(30 / 5) % 4 + 2`

b) `(17 - 11) / 3 + 3`

c) Rewrite part b, adding parentheses so the addition is performed before the division.

d) What is the value of the expression in part c ?

4) Write a program that does the following:

Step 1) Assign the number the user entered as a command line argument to an integer variable.

Step 2) If the number is less than 100, print "XX is small", where XX is filled in by the number the user typed.

Step 3) Otherwise, write "XX is large", with XX filled in as appropriate.

5) Given the variables and values:

```
boolean x = true;
boolean y = false;
int z = 5;
```

What is the value of each of the following Java expressions?

i) `x && (y || (z>1))`

ii) `!x || !(y && (z >= 5))`

iii) `y || (!x && !(z < 10))`

6) The following statement does not work as intended. Show what would be printed when it is executed assuming the user enters "5" when prompted, then fix the switch statement so that it will work properly. Also modify the program so that it prints an appropriate error message if the user types an invalid number.

```
StdOut.print("What day? ");
int day = StdIn.readInt();

if (day == 1)
      StdOut.println("Sunday");
if (day == 2)
      StdOut.println("Monday");
if (day == 3)
      StdOut.println("Tuesday");
if (day == 4)
      StdOut.println("Wednesday");
if (day == 5)
      StdOut.println("Thursday");
if (day == 6)
      StdOut.println("Friday");
else
      StdOut.println("Saturday");
```

7) Consider the following while loop:

```
int sum = 0;
int count = 5;
while (count <= 10) {
     sum = sum + count;
     count ++;
}
```

a) Before the loop, sum and count have been assigned values 0 and 5, respectively. Fill in the following table showing what values are assigned to these variables *during each iteration* of the loop (there may be more rows than you need in the table).

sum	count
0	5

b) Write a for-loop that does the equivalent to the while loop in 7a.

8) Consider the following Java source code:

```
int a = 4;
int b = 3;
double c;

c = a/b;
```

a) What is the value of variable c after executing the four lines of code above?

b) Rewrite the 4th line above using type-casts to make Java perform floating point division.

9) Write a snippet of Java code that compares two integer arrays (A and B) and outputs true if the arrays are "equal" (for example, if the array A contained three integers, 5 in the first position, 10 in the second position, and 15 in the third position, and the array B contained the same numbers in the same positions, the arrays would be equal, otherwise they would not be equal.)

10) Write a conditional that accomplishes the following:

i) If hair is 'S' and height is greater than 70 inches, invite the user to tryout for the blue team.

ii) If hair is 'C' and height is greater than 75 inches, invite the user to tryout for the green team.

iii) Otherwise, print out "Cold colors may not be for you."

Start with this code:

```
StdOut.print("Enter your height in inches: ");
int height = stdIn.readInt();
StdOut.print("Enter your hair style (S/C): ");
char hairStyle = (stdIn.readString()).charAt(0);
// write your conditional(s) here.
```

11) Rewrite your code from question 10 so that it accepts both upper and lower case letters.

12) Write a snippet of code that compares a `String` variable named `color` and assigns either `true` or `false` to a variable, `isPrimary`. If the `color` is the `String` “`red`”, “`green`”, or “`blue`”, the variable should be assigned `true`. Otherwise, it should be assigned `false`.

13) Write a snippet of code that assigns a double variable `PartialHarmonicSum` in the following way. It should use one integer variable named *n*, and a loop to calculate the *nth* harmonic number (the sum of the reciprocals of the first *n* natural numbers), what a mathematician would write as: `1 + 1/2 + 1/3 + … + 1/n`. You can assume that `n` is not negative.

14) Write a program that generates a table as shown below output. Avoid redundancy in your code. The first column is self explanatory. The second column is the square of the number in the first column. And the third column is the sum of all the squares up to and including that row of the table. Don't worry about aligning the columns perfectly.

```
N   N^2   sum
0    0     0
1    1     1
2    4     5
3    9    14
4   16    30
5   25    55
6   36    91
7   49   140
8   64   204
9   81   285
```

15) Write code that uses command-line input for two numbers, a width and a height, then prints a rectangle of plus-signs of the given size.

16) Write a snippet of code that will calculate and print the average of all the values stored in an array named `grades`. You can assume that `grades` is an array of `double` values that has already been initialized and contains at least one value.

17) Write a snippet of code that will calculate and print the minimum of all the values stored in an array named `grades`. You can assume that `grades` is an array of `double` values that has already been initialized and contains at least one value.

18) Write a snippet of code that will calculate and print how many B's a student received. You can assume the student's grades are stored in an array named `grades`, which is is an array of `double` values that has already been initialized and contains at least one value. Recall that a B is any grade between 80 and 90 (but not including 90).

19) Write a program that asks the user to input 4 integers. Store these into an array. Then print the numbers the user typed, but in reverse order. For example, if the user enters "2 4 6 8", then the program should print "8 6 4 2".

20) Show what values are stored in the array named `z` at the end of this snippet of code.

```
int[] x = { 2, 4, 6, 8 };
int[] z = { 5, 10, 15, 20 };
for (int i = 0; i < x.length; i++) {
    if (i % 2 == 0)
        z[i] += x[i] + z[i+1];
    else
        z[i] = x[i] + z[i-1];
}
```

21) Write a snippet of Java code that interactively asks the user to enter three integers and stores these into three variables. After the user has entered all three numbers, the program should print the three numbers in increasing sorted order. Hint: you will need several if/else statements (possibly nested, or in sequence).

22) An array of doubles contains audio data for some song. These are simply numbers, each one between -1.0 and +1.0, inclusive. To "normalize" the song volume to exactly 50%, find the number in the array with the largest absolute value, then divide every number by twice that amount. For example, if the largest number in the array is 0.3, then every number in the array should be divided by 0.6. Or, if the largest number (considering only the absolute values) was -1.0, the every number should get divided by -2.0. This procedure should guarantee that the song data will all fall neatly within the range -0.5 to +0.5.

23) The user inputs a number N on the command line, then create an array of length N and fill it with random numbers between 0 and 9, inclusive. Next, the program should determine *which number was chosen most often, and how often it was chosen*.

For example:

```
java MostOften 20
Here are 20 random numbers: 1 6 8 9 6 3 2 5
   7 9 6 6 2 8 5 5 4 3 8 1
Number 6 was chosen most, appearing 4 times.
```

24) You are given an array of strings, candidates. For example:

```
String[] candidates = {"Alice", "Fatima", "Diego", "Itaru"};
```

Write code that asks the user to vote for a candidate, repeatedly, until the user types "end". Then, print out the vote total for each candidate and the name of the winning candidate.

For example:

```
java Election
Candidates are: Alice Fatima Diego Itaru
Who do you vote for? Fatima
Who do you vote for? Itaru
Who do you vote for? Fatima
    ... much deleted interaction ...
Who do you vote for? Alice
Who do you vote for? Diego
Who do you vote for? end
Alice got 4   Fatima got 8   Diego got 10   Itaru got 5
The winner is Diego
```

25) You are given an array of strings, food, and an array of integers, calories. For example:

```
String[] food = {"Soup", "Apple", "Juice", "Donut"};
int[] calories = { 350,  130,   200,    1000 };
```

Write code that asks the user to enter how many servings they want for each, then prints out a nice summary of the total calories in their meal.

For example:

```
java MealPlanner
How many servings of Soup? 2
How many servings of Apple? 1
How many servings of Juice? 0
How many servings of Donut? 4
You ordered:
2 Soup... 700 calories
1 Apple... 130 calories
0 Juice... 0 calories
4 Donut... 4000 calories
Total is 4830 calories
```