CS 201 Final, Fall 2015, Monday-Wednesday Section

Read the questions carefully. One question is repeated from the sample exam, but several others differ in important ways from similar questions you have seen before.

**Part 1: Short Answers, 4 points each**

Answer briefly. Your answers must be clear and complete and must not be copied from any source.

1. Which of the following lines of code call static methods?

a) c.myMethod();

b) String[] myStrings = oldString.split();

c) Boolean b = input.nextBoolean(); // input is a Scanner that has already been intitialized

d) int x = Integer.valueOf(numString);

2. Consider the method main(String[] args) found in many Java classes.

a) Why is main always static?

b) What is the data type of args, the parameter taken by main?

c) Why does main take that particular data type as a parameter?

d) What happens when main() finishes executing?

3. What do the following lines of code or code snippets do?

a) boolean similarGrade = Math.abs(joesGrade-marysGrade) < 5;

b) if(string1.equals(string2)) System.out.println("same!");

c) x \*= y;

d)

String[][] myArray = new String[10][10];

for(int row = 0; row < myArray.length; row++){

for(int column = 0; column < myArray[row].length; column++){

myArray[row][column] = row + " " + column;

}

}

4. What is wrong with each of the following code snippets? Do not copy and paste error messages, explain the problem in your own words.

a) if(x == 3.14159) System.out.println("x = PI!");

b) double myDouble = JOptionPane.showInputDialog("Please enter your height in CM");

c)

public static String[] getStringArrayFromInput(int n) {

String[] theArray = new String[n];

for (int counter = 1; counter < n; counter++)

theArray[counter] = JOptionPane.showInputDialog(null,

"Please enter a String value");

return theArray;

}

c)

public static int getIntArrayFromInput(int count) {

int[] theArray = new int[count];

for (int i = 0; i < count; i++)

theArray = Integer.parseInt(JOptionPane.showInputDialog(null, "Please enter an integer value"));

}

return theArray;

}

5. a) Consider the following code. How many times will the System.out.println() be executed ?

int counter = 0;

do{

System.out.println("watch out for Godzilla!");

counter++;

} while(counter <= 100);

b) What about the System.out.println() in this one?

int counter = 0;

while(counter >= 0){

System.out.println("watch out for Godzilla!");

counter++;

}

6. Why does Java use "pass reference by value" instead of plain "pass by value" to pass arrays to methods as parameters and to return arrays from methods?

7. In Mooseville, where it is usually cold outside, people measure temperature in degrees below room temperature, so that 10 degrees is colder than zero, and zero is colder than -10. In Lizardburgh, where it is usually hot, people measure temperature in degrees above room temperature, so that 10 degrees is warmer than 0, which is warmer than -10. One Mooseville degree is equal in magnitude to two Lizardburgh degrees, and the standard of room temperature is ten Lizardburgh degrees warmer in Lizardburgh than in Mooseville. Thus, for example,

0 degrees M = -10 degrees L

-5 degrees M = 0 degrees L

-10 degrees M = 10 degrees L

-1 degree M = -8 degrees L

0.5 degrees M = -11 degrees L

1 degree M = -12 degrees L

Write one line of code that takes a double value M, which represents a Mooseville temperature, and converts it to a Lizardburgh temperature. I recommend you test your solution using the values shown above.

8. Write several lines of code that indicate whether an array of String contains the String "Godzilla". *Think carefully about how to test this.* The output can consist of simply using System.out.println() to print “yes” or “no.”

9. Suppose you have a two-dimensional array called myArray. What are the indices of the element in the third row,

fifth column? Be sure to list the indices in the correct order. Here is an example of a wrong but correctly formatted answer:

myArray[20][6]

10. Write one line of code that outputs whether or not the Unicode value for +, the addition symbol, is greater than the one for -, the hyphen.

**Part 2, One-Paragraph Answer (10 points)**

Answer the question in one well-constructed paragraph. A good paragraph contains a topic sentence, several

sentences that make specific point or give examples, and a concluding sentence.

Why do we break our applications into methods instead of writing everything in main()?

**Part 3, Programming Problem, 50 points**

Write a Java application that meets these requirements:

The method getLongArrayFromInput() takes an int called count as its only parameter. It uses console I/O to get an array of user-supplied longs of length count, then returns a reference to the array. For example, getLongArrayFromInput(10) would return an array of ten longs taken from user input. Use the Scanner method nextLong(), which is analogous to nextInt(). Hint: when taking input, use a for loop with a loop counter, not an enhanced for loop.

The method show32BitCount() takes an array of longs as its parameter and prints out the number of values in the array that could have been represented with ints (that is, the number of values that are at least -2147483648 but not greater than 2147483647.) This method should not return anything. The output should look something like this:

Array contains 10 long values, of which 4 are within the range of the Java int type

The method extractIntArray() takes an array of longs as its parameter and returns a **new array** of ints. For each case in which the long value is too large to be represented by the int data type, the new array contains the value

2147483647, and for each case in which the long was too small, the int array contains the value -2147483648. Hint: for the values that do fall within the range of ints, use a cast, ie (int) myArray[counter];

main() calls these methods in the appropriate order to show that they work correctly. To show that extractIntArray() is correct, create the original array, call the method, then print out both arrays separately, \*after\* the method is run.