Name: UTA ID:

Instructions:

1. The test is worth 100 points. The point value of each question is given with the question. There are also extra credit questions at the end.
2. The test is open book and open notes for all printed and hand-written material. You may NOT bring an electronic book or any electronic device to use during the test (no computer, no smart phone, etc.) You may use as much printed or written material as desired including copies of code examples.
3. You will write your answers on the test pages. If additional space is needed, you may use the back of the pages. Please make a note on the test page whenever your answer continues onto another page and indicate where the answer is.
4. Please write legibly. Your writing should readable if the test is sitting on a desk in front of me. I am not looking for perfect handwriting but it does need to be legible. I will deduct points if your answers are much more difficult to read than those of the general student.
5. If you have a question during the test, please raise your hand. The TA and I will be available to come hear your question. Sometimes we may not be able to answer your question because it gives you too much information but you should always ask.
6. You have 1 hour and 20 minutes to complete the test.
7. Assume we want to create a program that will iterate through a list of integers starting with 0 and ending with a value called MAX. On each iteration, the program should be adding the current integer to a running sum. The current total sum will be called currentSum. The program then, on each iteration, should also take the currentSum and the previous sum, i.e. the previous value of currentSum, add those two values together, and create a value called sumOfSums. Finally, in each iteration, print the current integer value, the current value of currentSum, and the resulting sumOfSums value, all on one line. The program should produce MAX plus one lines of output. The initial value of currentSum should be 0 and the initial value of sumOfSums should be 0.

1.a) Write the algorithm above in PSEUDOCODE. Lay out the algorithm in sentences with indentation. You should write a short sentence in words or phrases for each line that you think will be a Java statement. You do not need to write pseudocode for the class definition and the main method header but you should define any variables that you use that are not obvious from the program description above. {8 points}

1.b) What data types are you using for your variables and why? {4 pts}

1.c) Consider what loop control structure that you would choose to implement your algorithm and think about what variable(s) you will use to help control the loop. State which loop control structure, what test condition, and which variable(s) you would use for this algorithm AND explain why you choose this loop control, test condition, and variable(s). {6 pts}

1.d) If MAX is defined to be 8, write the output that should be produced by the program. {15 pts}

1.e) What mathematical pattern do you see in the sumOfSums column of values? {4 pts}

1.f) Write the Java code to implement the algorithm using your pseudocode from answer 1.a. Include the variable declarations then just write the Java loop for the algorithm. Define MAX as 50 in your program. You do not need to write the class declaration or the main method header. {15 pts}

1. Use the Java program T1Q2 to answer the following questions:

2.a) If *abool* is true, what does that tell you about *a*? {4 pts}

2.b) What do *bbool* and *cbool* tell us about their associated variables? {6 pts}

2.c) What output will you get for the input of *24 17 -2* ? Show the complete output with approximate spacing. {6 pts}

2.d) Is the program T1Q2 well-written? If so, explain why you think the code is good. If not, explain why you think the code is bad. Your explanation should be three sentences or less. {6 pts}

1. Use the Java program T1Q3 to answer the following questions:

3.a) How many conditions are tested using *a, b,* and *c* starting at *if(a < b)* until the end of the program if the input *99 14 38* ? Include *if (a < b)* as the first condition. {4 pts}

3.b) Does the program always test the same number of conditions regardless of the input values? Explain your answer. {6 pts}

3.c) Look at the code and decide what the values of *x, y,* and *z* will show when the program ends. Describe what the values will tell us rather than giving any specific number values. [HINT: Try two or three different sets of input test data to help determine the program purpose.] {5 pts}

4.a) Assume that a Scanner called *input* has been declared and is ready to use. Using the variables declared below, write a set of commands that would read one word followed by an integer then a second word followed by an integer then a floating point number and ending with a string until the end of the line. Here is a sample line of input to read: {6 pts}

*Thursday 1 October 2015 32.77 is the expected high temperature in Celcius today*

String dayOfWeek;

String month;

String message;

int day;

int year;

double temperature;

4.b) Using the variable *month* above, extract a substring that has the 3 letter abbreviation for the month name and store the abbreviation in a new String variable that you declare. {5 pts}

Extra Credit

XC1) Thinking about T1Q3 again, if you had *a, b, c,* and *d*, (four inputs instead of three) roughly how many conditions do you think you would need to test to correctly assign *w, x, y,* and *z*? Give your answer as a specific number of conditions or give it in reference to the count from 2.f, i.e. “Ten times as many conditions as found in question 3.a” {4 pts}

XC2) Use the declarations for *a, b, c,* and *d* to fill in the blanks below with *true* or *false*: (You may use T and F) {1 pt each; 4 total}

int a = 5; 1. c > a

int b = a \* 2; 2. d % (ceil(c)) == 4

double c = b \* 4 / 7; 3. (b >= (c / 5)) && ((d / a) > (b \* c))

double d = Math.pow(a, 3); 4. true || (a > b)

XC3) Write a two-line rhyme about learning Java. {Any answer will receive 2 points}