Name: Key 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. For multiple choice questions, circle the letter of your answer choice. Circle only one choice for your answer.
6. 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.
7. You have 1 hour and 20 minutes to complete the test.

1. Declare two arrays to hold information about people that you might give gifts to on the next gift-giving occasion like Christmas or Hanukkah.

The first array will hold single word names of recipients and their relationship to you, i.e. first\_name last\_name relation; examples would be

Joe De\_La\_Cruz cousin

Preethi O'Rourke friend

The second array of numbers will hold a) gift budget for recipient, b) the last year that you gave a gift to that person or 0 if you have never given that person a gift before, and c) a ranking of how important it is to you to give that person a gift with 1 = must give a gift, 2 = should give a gift, 3 = would like to give a gift, 4 = will give a gift if circumstances require it, 5 = other ; for example,

20.10 1996 5

25.00 2016 3

You should make the arrays big enough to hold 50 people. Write your two declarations below. Before your array declarations, you should declare constants to use to represent your columns.

// Constants (12 pts)

int MAXPEOPLE = 50; // can have int MAXCOLUMNS = 3 if desired

final int FIRST = 0;

final int LAST = 1;

final int RELATION = 2;

final int BUDGET = 0;

final int YEAR = 1;

final int RANK = 2;

final int CURRENTYEAR = 2016;

// Two dimensional array declarations

String[][]recipients = new String[MAXPEOPLE][3]; // if have MAXCOLUMNS then use it

double[][]recipsData = new double[MAXPEOPLE][3];

2) Define and give three examples of errors that could occur in the input data of the names and then define and give three examples of errors that could occur in the numeric data from question 1. You do not have to use the exact examples from question 1 but you must assume that the correct input formats of data look like the examples in question 1. The errors could be in data that is read from a file or data that is typed in by a user. (24 pts)

Names file possible errors:

Too many words – invalid length of relationship (1) or incorrectly reads 4th and after words as next line (2) – logic error, possible execution error

Too few words – does not have a relationship (3) or incorrectly reads next line as relationship (4) – logic error, possible execution error

[Spelling errors, numbers instead of words, actually don’t cause any problem]

Numbers file:

First value is not a number(1) – input mismatch – execution error

Second(2) or third value (3)being doubles or not numbers – input mismatch – execution error

Insufficient numbers on a line - incorrectly reads next line as for a number (4 – same as 3 above)

Too many numbers on a line – incorrectly reads 4th number as new budget, etc. (5 – as 1 above)

3) The data for the two arrays in question 1 comes from two files; *recipients.txt* and *recipientsData.txt*. There are the same number of lines of data in each file and the data on line X in one file is for the same person as the data on line X in the other file. Be sure to read the corresponding line data into corresponding rows in the arrays.

The format of the *recipients.txt* file is:

**first\_name last\_name relationship**

on each line of the file. Notice that there should only be blanks between the first name and last name and between the last name and relationship. Therefore if the person has a name with multiple words, like "Mary Anne" or "von Trapp" that these names will be written as Mary\_Anne and von\_Trapp in the file.

The format of *recipientsData.txt* file is:

**budget year rank**

on each line where budget is a positive amount of money in dollars and cents, year is a 4 digit positive integer less than the current year, and rank is a value between 1 and 5 inclusive.

Given the Scanner variables below, write the statements needed to read all the lines of the file into the arrays you declared in question 1. Do not do any error checking in this question – just write the Java needed to read all the data in the two files into the two arrays. (20 pts)

Scanner inPeople; // connects to the "recipients.txt" file

Scanner inData; // connects to the "recipientsData.txt" file

// You do not need to write any lines to set up the Scanners. You can assume they are set up

// to read from the listed files

String oneLine; int row = 0;

Scanner person; Scanner personData;

boolean inputValid = true; double tempDouble = 0;

int tempInt = 0;

while (inPeople.hasNextLine() && inData.hasNextLine()) // data still in the files

{

oneLine = inPeople.nextLine();

person = new Scanner(oneLine); // take one line out of file to read from

recipients[row][FIRST] = person.next(); // read three words from file

recipients[row][LAST] = person.next(); // No error checking as directed in the question

recipients[row][RELATION] = person.next();

oneLine = inData.nextLine();

personData = new Scanner(oneLine);mm // take one line out of file to read from

tempDouble = personData.nextDouble(); // or (because there’s no error checking…)

recipsData[row][BUDGET] = tempDouble; // recipsData[row][BUDGET] = personData.nextDouble();

recipsData[row][YEAR] = personData.nextInt();

recipsData[row][RANK] = personData.nextInt();

row++;

}

4) Given the definitions of the budget, year, and rank from questions 1 & 3, write the test condition that would be needed to validate each of the following items: (~~3~~5 pts each; 15 points total)

4.a) Budget: if (recipsData[row][BUDGET]>0) // or if (tempDouble > 0) w/ deduction

4.b) Year: if((recipsData[row][YEAR] >= 0) && (recipsData[row][YEAR] < 2016)) // or tempInt

4.c) Rank: if((recipsData[row][RANK] >= 1) && (recipsData[row][RANK] <=5)) // or tempInt

Given the code fragment below, answer the following questions, assuming that *recips* has Strings in it and *recData* has doubles in it.

String sp = " ";

if (recData[row][RANK] != 0){

System.out.println(row+" With rank "+(int)recData[row][RANK]

+" I have budgeted "+recData[row][BUDGET]

+" for "+recips[row][RELATION]+sp

+ recips[row][FIRST]+sp+recips[row][LAST]

+(((int)recData[row][YEAR] == 0)

?(" who has never received a gift from me")

:(" last gifted in "+(int)recData[row][YEAR]))

); }

5) Which of the following could be a line output by the code fragment? (5 pts)

a. 10 With rank 3 I have budgeted 30.0 for friend Duha Awayes last gifted in 1992

b. With rank 3 I have budgeted 25.0 for friend Preethi O'Rourke who has never received a gift from me

c. 4 With rank 1 I have budgeted 85.75 for sister Dorothea Feraci Klett last gifted in 2015

d. With rank 3 I have budgeted 35.25 for () Sean Fierce last gifted in 2013

6) What is the purpose of (int)recData[row][RANK] in the code fragment? (5 pts)

a. to print the ranking of this person against all the other people

b. to print the integer ranking as a double

c. to print the double ranking as an integer

d. to rank the quality of the row

7) If the year value is 0, what is printed at the end of the message by the code fragment above?

a. last gifted in 0 (5 pts)

b. last gifted in 2016

c. who has never given me a gift

d. who has never received a gift from me

8) Explain in words what the code below is doing. Don’t write pseudocode and don’t write each line in English. Look at the whole method and determine what it is accomplishing and describe that. Be specific about what it is doing and how. (14 pts)

public static void Action (double[][] data1, String[][] data2, int num, int num2)

{ int counter1 = 1;

int counter2 = (data1.length)-1;

String[] temp1 = new String[num];

double[] temp2 = new double[num];

int a = 0;

while ((counter1 > 0)&&(a < data1.length))

{ counter1 = 0;

for (int b = 0; b < counter2 ; b++)

{ if ((data1[b][num2] < data1[b+1][num2]) && (data1[b+1][num2] != 0))

{ counter1++;

temp1 = data2[b];

data2[b] = data2[b+1];

temp2 = data1[b];

data2[b+1] = temp1;

data1[b] = data1[b+1];

data1[b+1] = temp2;

}

}

a++;

counter2--;

}

}

The code above is

Performing a bubble sort on array *data1* and *data2* based on the value in column *num2* of *data1* as long as the value in column *num2* is not 0. The data is sorted from largest to smallest. When *data1* is sorted, the same row in *data2* is also sorted. The bubble sort is improved by shortening the number of comparisons each iteration and by ending the sort when a pass has no swaps.

[The underlined things are the primary things I was looking for in the answer.]

Extra Credit

XC1) Assume that you want to print the data in the arrays from question 1 in a table. Write an output statement (starting with System.out…) that prints all of the data for one person on row *index* in a table form. {4 points}

(something like this – formatted to make columns)

System.out.printf("%6d%15s%15s%15s%8d%8d%10.2f\n",row+1,

recips[row][FIRST],recips[row][LAST],

recips[row][RELATION], (int)recData[row][RANK],

(int)recData[row][YEAR],recData[row][BUDGET]);

XC2) If the input data is read from two files, one for words and one for numbers, which of the following lines would NOT be valid for the files described in question 3. {2 points}

a. Santa Claus imaginary\_friend

b. 15.93 2005 2

c. 23.05 2016 5 // Year can’t be 2016

d. Kris Kringle jolly\_old\_elf

XC3) Assuming we have an integer array called *data*, which of the following would NOT stop the program from compiling and NOT cause the program to crash? {2 points}

a. for (int index=0; index <= data.length; index++){ data[index] = index+data[index];}//OOB

b. j=0;do{ data[j] = data[j]+data[j+1];j++;}while(j < data.length–1);

c. if {data[n] < 0} (data[n] = -data[n];) // punctuation

d. while(k < data.length){ data[k] = Math.sqrt(data[k]); k++;} //double

XC4) Write a Java secret, real or made up. Be funny if possible. ☺{Any answer will receive 2 pts}