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. 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. Read through the questions before starting to work on any particular question. Then start working on the question you feel most comfortable with. Try to filter out the unneeded information in the problem before you work on it. Keep track of time and don’t spend too much time on any one problem.
7. You have 1 hour and 20 minutes to complete the test.

1. Assume that you have to read data from the user as follows: A team number, a one word team name, an amount of money that is the team budget, a number of team members, and the name of the team’s organization which could be multiple words. This input is entered from the file.

int teamNumber;

String teamName;

double teamBudget;

int teamMemberCount;

String teamOrganization;

Using the declarations given above, do the following:

1.a) Declare a Scanner object that will read from a file named *“teamInfo.txt”.* Declare any other variables needed for this. Assume that the file has *MAXLINES* or less in it to be read and each line has all five values in the order given above. Make sure to do any error checking that Java and NetBeans will require. Write the statements needed to read in the data and store it in the variables declared above. Read in the data assuming that the data is error free and that you are reading in each piece of data as its specific data type. Make sure that your lines of code will read all the data from the file. {30 points }

File inFile = new File(“teamInfo.txt”);

Scanner inData;

try

{

inData = new Scanner(inFile);

}

catch (FileNotFoundException fnf)

{

System.out.println(“No file found”);

inData = new Scanner(System.in);

}

while (inData.hasNextLine() )

{

teamNumber = inData.nextInt();

teamName = inData.next();

teamBudget = inData.nextDouble();

teamMemberCount = inData.nextInt();

teamOrganization = inData.nextLine();

}

1.b) After you have read one line of data, write the code that would print the values of those variables in the format below:

Team *teamNumber*, *teamName* is sponsored by *teamOrganization* with *teamMemberCount* members and budget of *teamBudget*

E.g. If *teamNumber* is 140, *teamName* “Cats on Fire”, *teamOrganization* is “Fluffy Farms”, *teamMemberCount* is 10 and *teamBudget* is 425, then your program should print: {15 points}

Team 140, Cats on Fire is sponsored by Fluffy Farms with 10 members and budget of 425

System.out.println(“Team ”+ teamNumber + “, “+ teamName +” is sponsored by “

+ teamOrganization +” with “+teamMemberCount

+” members and budget of “+teamBudget );

1.c.) Write the lines of code that would be needed to write output to a file named “teamInfoOutput.txt” . Then revise the output statement from 1.b.) so that the output would be printed to this file instead of to the screen. {20 points}

PrintWriter outFile = new PrintWriter(“teamInfoOutput.txt”);

outFile.println(“Team ”+ teamNumber + “, “+ teamName +” is sponsored by “

+ teamOrganization +” with “+teamMemberCount

+” members and budget of “+teamBudget );

2. Use the “Code for Test 3” page to answer the following questions:

2.a) What data is in the array that is being used in bSort and swap3? Describe the contents in words. (This is also related to what we have done in class.) {10 points}

The data in the array is the robot team data with team number, ranking points, and qualifying points info.

2.b) For the bSort, how will the list be ordered when the bSort method is complete? Describe the order the list will be in after sorting. {15 points}

The list will be ordered from smallest team number to largest team number

2.c) For line W, are all three of the variables declared on this line needed for this method or could fewer be used? Explain your answer. {10 points}

Fewer variables could be used. Only one temp variable is used at a time so only one is needed.

XC1) Declare an array that could hold 4 amounts of money in a row and 25 rows. {2 points}

double[][] money = new double[25][4];

XC2) Given 100 values in a file, use the Scanner below and read the values into the array you declared above. {3 points}

Scanner inputMoney = new Scanner(new File(“payday.txt”));

for (int row = 0; row < 25; row++)

for (int col = 0; col < 4; col++)

money[row][col] = inputMoney.nextDouble();

XC3) If you have 100 monetary values, describe at least three different ways that you could sort the amounts of money in logical orders. {3 points}

Money can be sorted largest first, smallest first, even (or odd) amounts first, whole dollar amounts vs. fractional (dollar and cent) amounts, debts sorted first then income,

XC4) Write a line of Java code that makes a rhyme.

{Any answer will receive 2 points}