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. There are two sections where you may select 5 questions from a larger number of questions and are not required to answer all questions. If more questions are answered than needed, the first 5 questions that are answered will be those counted for the section.
5. 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.
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. 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.
8. You have 2 hours and 30 minutes to complete the test.

Indicate whether each statement is True (T) or False (F). Answer any 5 out of the 6 questions.

{2 points each; 10 total}

T A valid (syntactically correct) Java statement can be written without a semicolon

F Java automatically knows the dimensions of any array

T The symbol “!” before a variable means “not”

T The declaration *int apple pie;*  is an invalid Java declaration

F “100” is an integer

F The *try*-*catch* structure lets Java decide how to handle specified exceptions

Give a short answer for each of the following. Your answer should be no more than 3 sentences long. Answer any 5 of the 8 questions. If you answer others, mark the ones to ignore by putting an ‘X’ on the question and writing “Do Not Grade” on that answer. {4 points each; 20 points total}

1) What does “scope” mean in Java?

Where an identifier (variable) can be seen and used in a program

2) In your own words, explain what a method is. Do NOT copy an answer from some source.

A method is – a set of statements that accomplish a task and can be called by other a single name from one or more places in the program.

3) Give an example of when a “while” loop would be preferable to a “for” loop and explain why.

When the condition for a loop is a test that needs to be true or false, instead of a count.

Example: testing to see if there is more data in a file

4) In your own words, explain what an array is. Do NOT copy an answer from some source.

An array is a group of data items, all the same type, stored contiguously that are referenced by one name and indexed locations

5) You have printed to an output file but the file is empty. Explain what is the most likely problem?

Forget to close the file with the close statement

6) If *outfile* is defined as a PrintWriter, what is the difference between *system.out* and *outfile.out*?

outfile is connected to an output file and system.out is connected to the screen

7) In your own words, explain the use and meaning of the two symbols “=” and “==”. Do NOT copy an answer from some source.

“=” is the assignment operator. It stores a value on the right hand side into a variable listed on the jeft hand side

“==” is the operator to test for equality of two values. The test returns True or False

8) Given the name of the variables below, tell what data type you would use for that variable and why.

nameInput a String because it indicates it holds a name

rainingOutside a Boolean because it indicates a question about whether it is raining outside

A) Write a few lines of declarations and code that will print out the products of the multiplication tables from 1x1 up to 10x10. There should be ten values on each row and a total of 10 rows. The data should line up in columns. The output should start off as : {12 points}

1 2 3 4 5 …

2 4 6 8 …

3 6 9 …

…

for (int row = 1; row <= 10; row++)

{

for (int count = 1; count <= 10; count++)

System.out.printf(“%d\t”,count\*row);

System.out.println();

}

B) Write a few lines of code to take an array of integer values sorted from largest to smallest and print the even integer values from the list in order from smallest to largest. [This is NOT the even indices (indexes), it is the values in the list.] Use the declaration below for the initial sorted data array to use. Declare any additional variables if needed. {10 pts}

final int MAX = 187;

int[] sortedArray = new int[MAX];

// Assume that a bunch of random integers have been put into *sortedArray*

// Also assume that the random integers in *sortedArray* have been sorted from

// largest at index [0] to smallest at index [MAX-1]

for (int k = MAX-1; k >=0; k--)

{

if ((sortedArray[k] % 2) == 0)

System.out.println("Even value at ["+k+"] = "+sortedArray[k]);

}

C) Assume you have a file of data with a maximum of 30 rows of data. Each row in the file has a first name (with no blanks), a last name (with no blanks), and a floating point number representing a grade. Declare the variables that would be needed to hold all of the names and grades as they are read into your program from the file. Use the constant declared below as needed. {6 points}

final int MAX\_ROWS = 30;

String[][] names = new String[MAX\_ROWS][2];

double[] grade = new double[MAX\_ROWS];

// arrays are needed to hold the data on 30 lines in the file. At least two arrays are needed

// because there are two different types of data – Strings and floating point numbers

D) Given the code below,

final int SEMESTER\_SCORE = 0;

final int TEST1 = 1;

final int TEST2 = 2;

final int TEST3 = 3;

double[][] gradeArray = new double[MAXSTUDENTS][4]; // Assume MAXSTUDENTS is defined

// Assume gradeArray has data put in it somehow (could be from user or from a file)

// column 1 (TEST1) is score on Test 1, etc. for column 2 with TEST2 and col 3 with TEST3

// column 0 (SEMESTER\_SCORE) is calculated and stored in that column

// Assume numValuesInArray is declared and is the number of students/rows of data in array

for (pass = 0; (pass < numValuesInArray); pass++ )

{

for (index = 0; (index < numValuesInArray - 1); index++ )

{

if ( ) // line T

{

swap2Values(gradeArray, index, SEMESTER\_SCORE, TEST1, TEST2, TEST3);

}

}

}

Write the correct test that would be needed in the if statement at // *line T* in order for the array to be sorted from largest Test 3 score at index [0] to smallest Test 3 score. (You can write the test condition below here) {6 points}

gradeArray[index][TEST3] < gradeArray[index+1][TEST3]

E) Write the pseudocode to read in integers from a file, then print those values out in reverse order to how they were read in from the file. Assume there is one number per line of the file and that there are less than 100 lines in the file. {8 points}

Declare File and Scanner

Declare an array, of size 100 max, to hold the integer data from the file

While there is still more values in the file

Read a value from a line and store in the array

COUNT the number of lines/values in the file

Print the array from the last index [COUNT-1] to the first index [0]

F) Write the code that implements your pseudocode from question E. {10 points}

//Declare File and Scanner

File inInts = new File(“input.txt”);

Scanner inFile;

try

{ inFile = new Scanner(inInts); }

catch (FileNotFoundException fnf)

{ inFile = new Scanner(System.in); }

//Declare an array, of size 100 max, to hold the integer data from the file

int[] fileInts = new int[100];

//While there is still more values in the file

int count = 0;

while (inFile.hasNext())

{

//Read a value from a line and store in the array

//COUNT the number of lines/values in the file

fileInts[count++] = inFile.nextInt();

}

//Print the array from the last index [COUNT] to the first index [0]

System.out.println(“Reverse values from file”);

for( int j = count-1 ; j >= 0; j--)

System.out.println(fileInts[j]);

G) Given the File and Scanner definitions below, assume that the file called “ShoppingList.txt” has 75 lines or less and has two values on each line. The first value is a price for an item and the second value is an amount of that item. Declare a floating point array with 75 rows and 3 columns. Read in each line from the file and store the two values from that line into the first two columns of one row in the array. Then use the price and amount values to calculate a total amount and store that in the 3rd column. When all the data has been read from the file, print out the sum of all the totals as the total cost of the shopping list. {18 points}

File list = new File(“ShoppingList.txt”);

Scanner shopList = new Scanner(list);

final int PRICE = 0;

final int AMOUNT = 1;

final int TOTAL\_AMT = 2;

double[][] shoppingList = new double[75][3];

int row = 0;

double totalSum = 0;

while ((shopList.hasNextDouble()) && (row < 75))

{

shoppingList[row][PRICE] = shopList.nextDouble();

shoppingList[row][AMOUNT] = shopList.nextDouble();

shoppingList[row][TOTAL\_AMT] =

shoppingList[row][PRICE] \* shoppingList[row][AMOUNT];

totalSum += shoppingList[row][TOTAL\_AMT];

shopList.nextLine();

}

System.out.println();

System.out.println("Total sums in file = "+totalSum);

Extra Credit

XC1) *for* and *while* are examples of what kind of control structure? {2 points}

Repetition or iteration (2 pts)

Loops (1 pt)

XC2) List the items need to be imported from Java libraries in order to use Scanners and Files?

{2 points}

import java.io.File;

import java.io.FileNotFoundException;

import java.util.Scanner;

XC3) Given the name of the variables below, tell what data type you would use for that variable and why. {4 points}

volumeOfRoom a volume measurement would either be an integer or a floating point number and the floating point number would be safer - double

studentCount an integer – no partial people

XC4) What kind of cup does Dr. T usually bring her Coke to class in?

{Any answer will receive 2 points}

{2 extra points} List one of the sayings on one of Dr. T’s cups.