Name: Key All (version C) UTA ID:

Yellow is version A, Blue is version B, and White is version C

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.

1. Assume we want to create a program that will iterate through a set of grades entered by the user and then will average these grades. The program will need to sum up all the grades that are entered using a variable *totalGrades* and will need to count how many grades the user enters with a variable *countGrades*. The input variable for the grades should be called *nextGrade* and should be initialized to 0.

Your program should have a loop to allow the user to enter grades until *nextGrade* is -1. In the loop your program should start by asking the user to enter an integer grade between 0 and 100 or to enter -1 if they are finished entering grades. The program then reads the grade and assigns it to *nextGrade*. If *nextGrade* is between 0 and 100, then it is added to *totalGrades* and *countGrades* is incremented. Then the loop test is checked again.

When the loop test fails, print out a message stating the number of grades that were entered and print the average of all the grades. Q4 on B, Q3 on A

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. {11 points}

initialize nextGrade, countGrades, and totalGrades to 0

loop until the entered nextGrade is -1

Ask user to enter a grade and

Read nextGrade from user

Check if nextGrade is between 0 and 100 and if the grade is valid,

Add nextGrade to totalGrade

Increment countGrades

End of loop

Print countGrades and average calculated as totalGrade / countGrades

1.b) What data types are you using for the given variables and why? List the variable name and tell the data type and why you chose that data type. {6 pts}

double totalGrades; // a sum in case it got large, could be used as an average

int countGrades; // just a counter (integer)

int nextGrade; // a grade from 0 to 100

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}

loop control structure -- ‘while’ loop because there is a boolean condition not a counter

what test condition – check to see if nextGrade was -1

which variable(s) -- ‘nextGrade’ is the value the user enters

1.d) If the user enters the values 59, 87, 60, 91, -1, write the output that should be produced by the program. {5 pts}

59

87

60

91

-----

297 / 4 = 74.25

count of 4 and average is 74.25

1.e) 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. Include a scanner to read in a value for *nextGrade* in your program and include the output statements needed. You do not need to write the class declaration or the main method header. Q4b on B {15 pts}

// initialize nextGrade, countGrades, and totalGrades to 0

double totalGrades = 0;

int countGrades = 0;

int nextGrade = 0;

Scanner input = new Scanner(System.in);

// loop until the entered nextGrade is -1

while (nextGrade != -1)

{

// Ask user to enter a grade and

System.out.println(“Please enter a grade between 0 and 100, or enter -1 to end”);

// Read nextGrade from user

nextGrade = input.nextInt();

// Check if nextGrade is between 0 and 100 and if the grade is valid,

if ((nextGrade >= 0) && (nextGrade <= 100))

{

// Add nextGrade to totalGrade

totalGrade += nextGrade;

// Increment countGrades

countGrades++;

}

// End of loop

}

// Print countGrades and average calculated as totalGrade / countGrades

System.out.println(“You entered “+countGrades+” grades with an average of “+(totalGrade / countGrades));

2. Use the Java program Test1Fall16Q to answer the following questions: Q1 on B; Q4 on A

2.a) Write the Java statements that would arithmetically get the month and day values from the input values of mmdd1 and mmdd2 {6 points}

mm1 = mmdd1/100; // Get the mm1 value from mmdd1

dd1 = mmdd1%100; // Get the dd1 value from mmdd1

mm2 = mmdd2/100; // Get the mm2 value from mmdd2

dd2 = mmdd2%100; // Get the dd2 value from mmdd2

2.b) What does the program Test1Fall16 do with the two inputs? {15 points}

The program calculates the number of days between mmdd1 and mmdd2

3. Given the possible variable names below, choose a reasonable data type and write a declaration, with data type to the left, for each of the variable names. Use primitive types wherever possible. Indicate to the right side of the name if it cannot be used and if not, why not. Q3 on B, Q2 on A {15 points total}

String (maybe int) *dayOfTheWeek*

*for Loop* has a space in the middle and one word is a keyword

int *numOfEntrees*

*2Long* starts with a numeric digit

String *capitalCity*

String, int, double, boolean *JELLO // pretty much anything will work*

*!me* starts with the ! symbol

4. Assume that the user has entered an integer called *monthValue* that represents a month, where 1 represents January, 2 is for February, … and 12 is for December, and that you are given the pseudocode below: Q2 on B, Q1 on A

String *season*;

String *msg*;

boolean *valid* = true;

If *monthValue* is valid and is equal to December, January, or February, //start pseudocode here

Then set *season to* “winter”

Otherwise If *monthValue* is valid and equals March through May,

set *season* to “spring”

else If *monthValue* is valid and equals June - August,

set *season* to “summer”

Otherwise If *monthValue* is valid and equals September, October, or November,

set *season* to “fall”

else set *season* to “year” and *valid* to false // end of the “season” if-else

If *monthValue* is December, March, June, or September,

Set msg to “beginning”

Otherwise if *monthValue* is Jan., Apr., July, or Oct.,

Set msg to “middle”

Else if *monthValue* is Feb., May, Aug., or Nov.,

Set msg to “end”

else set msg to “an invalid month” and *valid* to false

Finally print “Month “+*monthValue*+” is “+(valid?“the”:””) +*msg*+” of the “+*season*

4.a) Write a set of if, if-else, and nested if-else Java statements to implement the pseudocode above. You only need to write the lines of code to implement from the first if statement (with // comment) above through the print statement. (You don’t need to write the class statement and don’t rewrite the variables declared above.) Use the back of the previous page or of this page if more space is needed. {8 points}

// The code below is one way to implement the code but not the only way to do it.

int monthValue = 0;

// Scanner input = new Scanner(System.in);

// System.out.print("Enter an integer value representing a month : ");

// monthValue = input.nextInt();

Input Validation

if (((monthValue == 12) || (monthValue == 1)) || (monthValue == 2 )))

{ //Then set season to “winter”

season = "winter";

}

else if ((monthValue >= 3) && (monthValue <= 5))

{ //set season to “spring”

season = "spring";

}

else if ((monthValue >= 6) && (monthValue <= 8))

{ //set season to “summer”

season = "summer";

}

else if ((monthValue >= 9) && (monthValue <= 11))

{ //set season to “fall”

season = "fall";

}

else // month is < 1 or > 12

{

season = "year";

valid = false;

} Input Validation

if (((monthValue == 12) || (monthValue == 3)) ||

((monthValue == 6) || (monthValue == 9)))

{ //Set msg to “beginning”

msg = "beginning";

}

else if (((monthValue == 1) || (monthValue == 4)) ||

((monthValue == 7) || (monthValue == 10)))

{ //Set msg to “middle”

msg = "middle";

}

else if (((monthValue == 2) || (monthValue == 5)) ||

((monthValue == 8) || (monthValue == 11)))

{ //Set msg to “end”

msg = "end";

}

else

{

msg = "an invalid month";

}

System.out.println("Month "+monthValue+" is "+(valid?"the ":"")+msg+" of the "+season);

4.b) Rewrite your “season” if-else statements above as a Java switch statement. Do not rewrite any other part of the Java code. {8 points }

switch (monthValue)

{

case 1:

case 2: season = "winter";

break;

case 12: season = "winter";

break;

case 4:

case 5:

case 3: season = "spring";

break;

case 7:

case 8:

case 6: season = "summer";

break;

case 10:

case 11:

case 9: season = "fall";

break;

default: season = “year”;

break;

}

4.c) In the code in 4.a, identify the lines of code that you wrote to check the validity of the user’s input. Put the words “Input validation” on the right side of your page and use arrows to indicate the line or lines that are doing the input validation. {5 pts}

Extra Credit

XC1) List three categories of control structures in Java. {3 points}

Sequential structures

Selection structures

Repetition structures

XC2) int ab = 5;

while (ab < 10) {

// stuff

ab++;

}

Rewrite the while loop above as a for loop. {5 points}

for (in tab = 5; ab < 10; ab++)

{

//stuff

}

XC4) List one way the Java impacts your life. Be amusing. ☺ {Any answer will receive 2 pts}