**QUESTION PACKET**

CS-200, Programming I Final Exam

Saturday, August 10, 2019

8:30 a.m. – 10:30 a.m.

**Instructions:**

1. **Do not turn this page until told to do so.**

1. This exam is **closed book** and **closed notes**.
2. **Write your STUDENT ID Number on every page**. If you do not write your ID Number on a certain page, you **will not receive credit for the question** **on that page!**
3. **Do not write your name** **on any page (except this one)**.If you write your name on a certain page, **you will not receive credit for the question on that page!**
4. There are **five** problems on the exam, one per page. Each problem will be graded as "pass" or "fail". To pass this final exam, you must pass at least **three** out of the five questions. Once you start, verify you have all 5 problems. If you are missing any question(s), notify a proctor immediately.
5. You must place your answer to a question on the **specified** page in the **ANSWER PACKET**. If you place your answer anywhere else, **you will not receive credit for it!**
6. For problems that ask you to write a **method**, you must use the given method header **exactly** as shown, and you do not need to write the main( ) method.
7. You may use "kb", "kbd", "keyboard" or "input" to get input from the keyboard without defining them. Assume the following import statement and keyboard declarations have already been written for you (do not write these statements in your answers):

 import java.util.Scanner;

 Scanner keyboard = new Scanner (System.in);
 or Scanner kb = new Scanner (System.in);

 or Scanner kbd = new Scanner (System.in);

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

1. You may use "SOP" as an abbreviation for "System.out.print" and "SOPln" or "SOPL" for "System.out.println".
2. You do not need to do any error checking of input values, **unless the problem specifically asks you to do so!**
3. If at any time during the exam you are caught looking at other papers or communicating with other students in any way, you will receive an **F** for the course.

**Question 1**

The game “Chutes and Ladders” consists of a game board grid of 10 rows by 10 columns. In this board, cells are filled with numbers that are descending on even rows and ascending on odd rows

(See the board below).

Your task is to create the method createBoard() that returns an integer 2D array where each cell has a number, from 1 to 100, just as a depicted in the chutes and ladders’ board.

**Note**: You are not allowed to fill individual cells with hard coded statements (i.e. a[0][0]=100; ).

You do not have to print the array.

Just return it.

|  |  |
| --- | --- |
| mage result for chutes and ladders boardSample board. Notice the numbering of the cells. | 100 99 98 97 96 95 94 93 92 91 81 82 83 84 85 86 87 88 89 90 80 79 78 77 76 75 74 73 72 71 61 62 63 64 65 66 67 68 69 70 60 59 58 57 56 55 54 53 52 51 41 42 43 44 45 46 47 48 49 50 40 39 38 37 36 35 34 33 32 31 21 22 23 24 25 26 27 28 29 30 20 19 18 17 16 15 14 13 12 11 1 2 3 4 5 6 7 8 9 10 Print out of the desired output. |

**Question 2**

You have the first name, last name and phone number of all your friends stored in three arrays. One for first names (fnames), one for last names (lnames) and one for phone numbers (phones). Assume phone numbers are long integers. The elements of each array are aligned so that the ith element of each array describes the same person. See below

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | fnames |  |  | lnames |  |  | phones |
| 0 | John |  | 0 | Smith |  | 0 | 7734567654 |
| 1 | Alice |  | 1 | Cooper |  | 1 | 3124756898 |
| 2 | Cooper |  | 2 | Sell |  | 2 | 3126543212 |
| : | : |  | : | : |  | : | : |

You need to code the method

**void getNumber(String[] fnames, String[] lnames, long[]** **phones**) that asks the user for an area code and prints all the full names (first and last) and phone numbers of people whose phone number begin with a given area code. Assume the arrays fnames, lnames and phones are already created and populated. If the area code does not exist, nothing is printed.

Example 1: Given the arrays above, a sample

getNumber(String[] fnames, String[] lnames, long[] phones) should behave like this:

Area Code?

*you enter 312 and the computer prints:*

Alice Cooper 3124756898

Cooper Sell 3126543212

Example 2: If the arrays were filled as shown above, If you enter 847, the computer would not print anything.

Area Code?

*you enter 847 and the program ends without printing:*

**Question 3**

Your company wants an automatic check writer and they divide the job among software developers. Your job is to take numerical small numbers and convert them into the word spelling of those numbers. The range of numbers you have been given to create the words for are numbers between 20 and 99 inclusive (You are creating what goes in space #4 of a check).

 

You have to implement the method void spellNum(int num) as follows:

* For numbers less than 20 the program outputs “too low”
* For numbers greater than 99 your program outputs “too high”
* For numbers between 20 and 99 it should spell the number with a dash between the tens and units (e.g. eighty-five, fifty-three, etc.). Of course, when printing numbers that are divisible by ten, like twenty, fifty, eighty, etc. there should be no dash.

For example, if the input is 23 your method prints “twenty-three”; if the input is 84 your method prints “eighty-four”. If the input is “30” your method prints “thirty”. If the number is 19 your program prints “too low” and if the number is 100 your program prints “too high”

**Question 4**

You are going to help students sharpen their multiplication skills. Your job is to program a multiplication test generator.

Your program will ask the user how many multiplication problems to test. Assume they enter a positive integer value.

Then your program will present the user one problem at a time that will consist of the multiplication of two random integers between 1 and 100, and it will ask for the answer.

Recall, the method Math.random( ) returns a value greater or equal to 0.000 and less than 1.000.

Your program will track correct and incorrect answers and will display the number of correct and incorrect responses after the last problem is answered.

Sample run:

How many problems?

*User enters 4*

Problem 1: 34\*7=?

*User enters 238 (which is correct)*

Problem 2: 99\*75=?

*User enters 7425 (which is correct)*

Problem 3: 78\*10=?

*User enters 780 (which is correct)*

Problem 4: 1\*1=?

*User enters 5 (which is incorrect)*

Total Score: Correct: 3; Incorrect: 1

**Question 5**

The last message from the Opportunity –the last mars rover— was a bunch of numbers that conveyed, basically: “My batteries are low and it’s getting dark.” The new mars rover, Curiosity, needs to be smarter about battery usage.

You are in charge of estimating whether the Curiosity is ready to go on a mission on mars. The Curiosity charges its battery using solar power. Your job is to estimate whether the rover can accomplish a given round trip knowing the following:

1. The battery holds 45000 Watts when it is at 100%.
2. The battery will lose 10% of its actual charge due to normal operations regardless of how far the rover has to travel. However, in cold seasons it loses 20% of its current charge.
3. 150Watts is good for 1 mile.

You must program a function boolean canDoRoundTrip(int percent,int miles,boolean cold) that takes in three parameters: a reading of the percentage of battery charge in the rover as an integer; the miles you want it to travel in one direction as an integer (remember it is a round trip) and whether it is a cold season or not (boolean). This function must **print** the percentage of charge, the total miles for the round trip and the miles the rover can actually travel. Then, it should **return** true if the rover can complete the trip, false otherwise.

Sample runs:

Run 1

canDoRoundTrip(90,120,false)

*Prints:*

Percentage of Charge:90; KW:40500.0

Total Miles for round trip:240

Total Miles available:243.0

*Returns:*

true

Run 2:

canDoRoundTrip(90,120,true)

*Prints:*

Percentage of Charge:90; KW:40500.0

Total Miles for round trip:240

Total Miles available:216.0

*Returns:*

false