Sample Qualifying Exam for Direct Entry Track

This exam is posted for prospective students of the Master of Science in Computer Science Program at Maharishi University of Management (Computer Professionals Program). In order to qualify for the DIRECT ENTRY track, incoming students must pass an exam similar to the one given below upon arrival at the University. If students are not able to pass this exam, they may enter the PREPARATORY TRACK only if they pass the PREPARATORY TRACK EXAM. Actual exam questions will differ from those below. The sample exams are posted here in order to help prospective students assess their readiness for study in the program.

1. **[Tests problem solving and a little bit of Java language]** Write a Java method removeDuplicates that removes all duplicates in a given list. Assume the following:

a. The method accepts an object of type List

b. The return type of the method is void

c. Duplicates are determined using the equals() method (rather than by the == operator)

Your implementation of removeDuplicates should handle, in an appropriate way, the case in which a null List is passed in to the method.

Test your method by writing code in a main method, which does the following:

a. It creates an instance of List and loads it with the following String values:{"good", "better", "best", "best", "first", "last", "last", "last","good"}

b. It invokes the removeDuplicatesMethod, passing in this instance of List

c. It outputs the modified list to the console

2. **[Tests Java language and prob solving]** Write a Java method testForSum which determines whether a given array of integers contains three entries whose sum is equal to a given integer. Assume the following:

a. The method accepts an array intArr of int's and an int testInt as its two arguments

b. The return type of the method is boolean

c. The method returns true if and only if there are distinct integers i, j, k such that intArr[i] + intArr[j] + intArr[k] equals testInt.

Test your method in a main method, which passes the following input values

{5, 1, 23, 21, 17, 2, 3, 9, 12}, 22

into the method testForSum, and which outputs the return value to the console.

3. **[Tests knowledge of data structures]** Create your own linked list (do not use any of the classes provided in the Collections API). Implement the following two operations:

If you are using jdk1.4 or before:

void add(Object ob);

boolean find(Object ob);

String toString();

If you are using j2se5.0 and you know generic programming:

void add(T ob);

boolean find(T ob);

String toString()

The toString method should arrange the elements of the list in a comma-separated sequence, in the following format:

[elem0, elem1, elem2, …, elemN]

Test your linked list in a main method which does the following:

a. Creates an instance of your list and adds the following Strings to it:

"Straight", "Bent", "Equals", "Well", "Storm"

b. Uses your find function to search for the keys "Well" and "Strength"

c. Outputs both the input list and the search results to the consoleand output the results to the consoleby repeatedly using your add function to populate a new instance of your linked list with Strings, and then outputting to console the boolean result of searching for some String in this list.

4. **[Tests basic knowledge of recursion]** Write a recursive static Java method that accepts an array arr of integers argument returns a list of all permutations of these integers.

(A *permutation* of a sequence of integers is a re-arrangement of the integers. For example, one permutation of 1, 3, 4, 8, 2 is 3, 1, 2, 8, 4.) For this problem, you may assume that the input array contains no duplicate entries. Your method should return an ArrayList of int arrays.

Next, test your method using a main method; the main method should pass in the following array: [1, 5, 4, 2]; then, it should print to the console the resulting list of permutations.

5. **[Tests knowledge of concept of static]** Create a Java class that allows at most 5 instances of itself to be created. Call your class JustFive. Provide a main method in your class that attempts to create 6 instances of your class.

BACK TO TOP