# Introduction to Computers and Engineering Problem Solving

# Spring 2005

# Problem Set 10: Diagnostics

# Due: 12 noon, Session 35

## Problem summary

You are to write a program to assist users in troubleshooting printers. The program will print a set of questions and multiple choice answers from a file. It will display the first question in the list and prompt the user for input. Once the user chooses one of the multiple options, an answer will be suggested and/or another question will be asked or the program will state that it can't help with troubleshooting this problem. The program will write the answers given by the user to a file.

The file of questions is on the MIT server. Its format is:

```
0     What printer problem are you having?      1,2,3,4,5,6,7,8
1     Can't install local printer   9
2     Document doesn't print  14
3     Some printer features don't work    79
4     Printed text is incomplete    84
5     Fonts are missing 89
6     Printing is unusually slow    94
7     Message appears that "printer driver is not compatible"     99
8     Message appears that "printer driver not installed"   99
9     Remove old drivers; install new driver. Does this solve problem?
      10,11
10    Yes   12
11    No    13
12    Thank you for using the troubleshooter    -1
13    Troubleshooter unable to solve your problem     -1
14    Can you print another document from same program, or print test
page from Notepad?      15,16
…
```

There are three fields, delimited by tabs (\t): the message number, the message to be displayed, and the additional message numbers to be displayed with the choices/questions to be answered to continue troubleshooting. For example, the program will start by displaying message 0, which then displays choices 1 through 8. The user will pick choice 1 through 8. If she picks 1, message 9 is displayed next, along with choices 10 and 11. If the user picks 10, message 12 is displayed and the program ends, since the additional message is -1, a value that indicates the program has ended.

Since this is the last homework of the term, you should be able to design the program yourself. The program must display the messages, which are in a general tree data structure, get user input as needed, and navigate through the tree until a leaf node is reached. At the end of the program, you must write the questions and user responses to a text file. The input may be via JOptionPanes; the output (the additional messages displayed from which the user must choose an answer) may be written using System.out.println() or may be included in the JOptionPane.. The files are read and written using streams appropriate to text files. You can assume that the message numbers start with 0, are in ascending order, and have no gaps. Catch exceptions on user input.

## Extra Credit

In addition to your solution above, you may implement a Swing GUI for this problem set and get up to 40 extra credit points. **You must first complete and submit the entire non-GUI solution as outlined above.** Do not attempt to develop your GUI until you have completed the normal assignment since it will be graded separately. You should understand that a GUI solution often requires changes to the rest of your code. Therefore we require you to develop your GUI solution in a separate directory (folder), copying all the files you need. When you submit your solution, first submit your original solution. Then upload your extra credit solution as a second .zip file. Both versions should contain all the files needed to compile and run your solution.

**You can get extra credit on only one homework from problem sets 8 to 10.**

## Turn In

1. Place a comment with your full name, MIT server username, section, TA name and assignment number at the beginning of all `.java` files in your solution.
2. Place all of the files in your solution in a single zip file.
    a. Do not turn in electronic or printed copies of compiled byte code (`.class` files) or backup source code (`.java~` files)
    b. Do not turn in printed copies of your solution.
3. Submit this single zip file.
4. Your solution is due at noon. Your uploaded files should have a timestamp of no later than noon on the due date.

## Penalties

- 30% off if you turn in your problem set after Friday noon but before noon on the following Monday. You have one no-penalty late submission per term for a turn-in after Friday noon and before Monday noon.
- No credit if you turn in your problem set after noon on the following Monday.