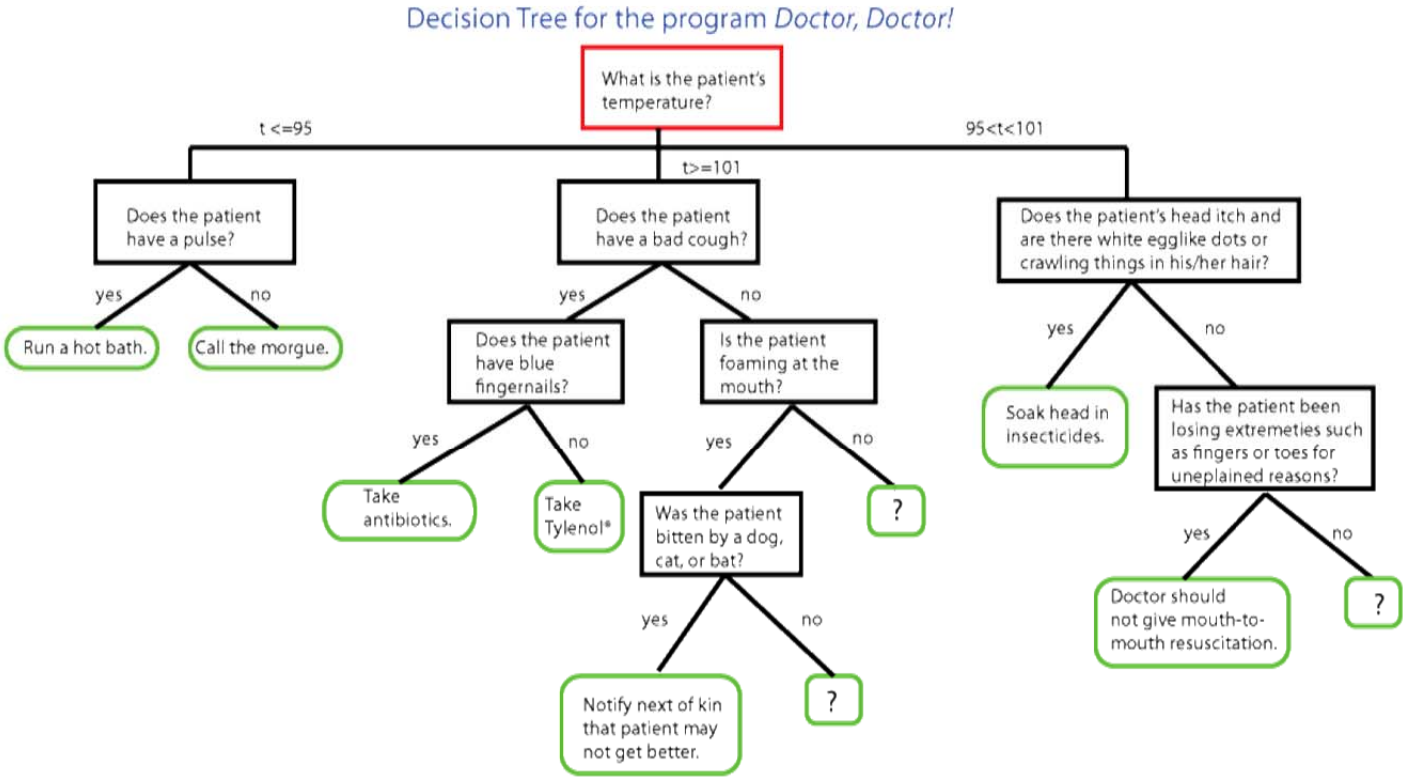
CS-141 Introduction to Programming
Writing an Expert System

Learning Objectives: Students will learn how to use nested if-else statements to implement a decision tree (the Expert System) in Java .

Example

The link (http://www.willamette.edu/~gorr/classes/cs141/lectures/doctor.htm) describes an exercise to implement a medical diagnosis "expert system" which uses a series of questions to arrive at a diagnosis. The structure of the rules given can be seen in what is called a decision tree:

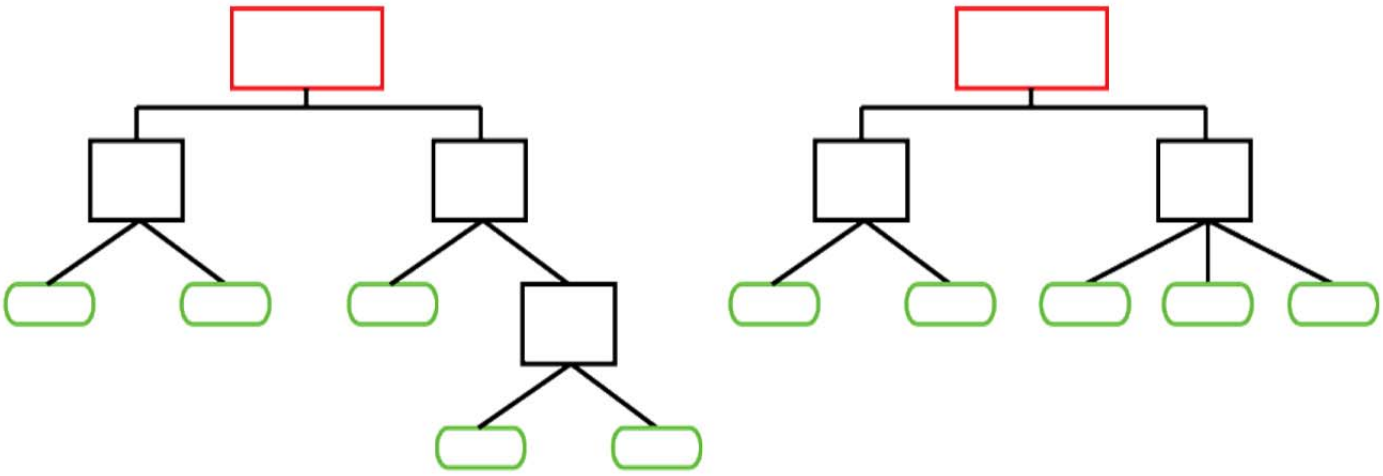


Each box is called a node. The top red node is called the root node; it is where the diagnosis begins. The green nodes at the bottom are called leaf nodes; they are where the final diagnosis can be found. Note that there are some missing leaves in the written rules. These are specified by "?" in the above tree. In these cases, it is assumed the doctor does not have a diagnosis; the program should let the user know this.

Assignment – Part 1: Software Implementation

Your assignment is as follows:

1. Pick a topic you know something about.
2. Carefully and neatly draw a decision tree. At a minimum, your decision tree should have at least two levels and 5 leaf nodes. For example, two possible structures are shown below:



3. Every decision should eventually lead to a response, i.e. there should be no missing leaf nodes (e.g. as in the "?" above). Also, try to be consistent in the structure, for example, in the tree Medical Diagnosis example above, the "yes" always appears on the left branch and the "no" on the right. This will make coding a bit easier.
4. Use your decision tree to guide the writing of your code. Don't implement everything at once. Start with the top most if-else statement (i.e. the one right below the root) and compile & test before continuing. Maintaining proper indentation will significantly increase the readability of your code and decrease how much time you spend having to debug.
5. When done, you should test your code by running it once for each possible "path" from the root to a leaf node.

Assignment – Part 2: Software Testing

Pick a partner in the class to work with:

1. Exchange copies of your decision trees.
2. Write your name on their decision tree paper indicating that you are the "Tester".
3. Run their code at least once for each possible output (leaf node). To run their code, they can either give you a copy of their code on a thumb drive, or you can swap seats and run their code on their computer. Each time you run their code, check off the corresponding leaf in their decision tree. This will help you keep track of what you have tested.
4. Based on the results of running their program, answer the questions on the next page.
5. When you are done, staple together all of the following: both copies of the decision tree, both sets of answers to the questions, and paper copies of the code (your instructor may also ask for you to submit an electronic copy of your code.)

CS-141 Introduction to Programming
Testing an Expert System

Your name: _____

Your partner's name (i.e. the name of the code's author): _____

The program's area of expertise: _____

1. Did your partner's program correctly implement their decision tree? If not, what went wrong?

2. Was their program understandable and easy to use? If not, what advice would you give your partner for improving their program?

3. Is there anything that you particularly liked about their program?