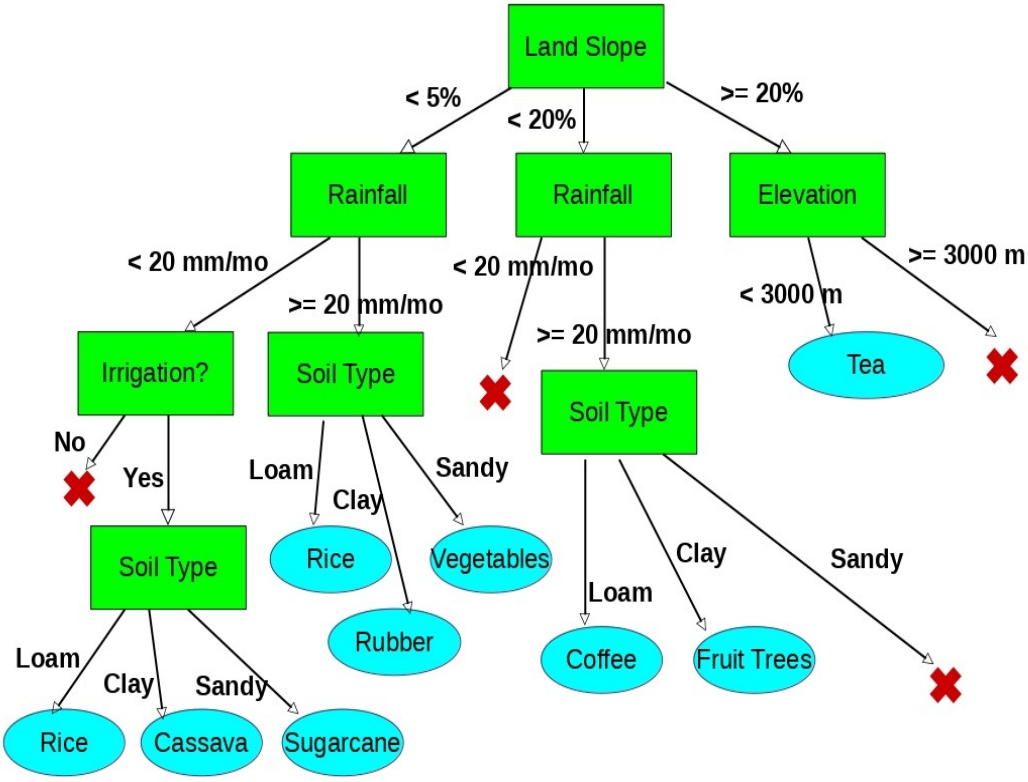
SEN-102 Introduction to Programming
Alternative Assessment Activity SEN-102:00010
Creating Conditional Structures
Last update: 22 December 2023

Assignment Instructions

Decision trees provide a method for making a choice or deciding on a category based on the attributes of particular instances of some entity. For instance, in AIC-501 we saw the following example of a decision tree that a farmer might use to decide what crop to plant, given information about the farmer's land such as the slope, amount of rainfall, and so on.



For this assessment, you must write a program in C called **`cropDecision.c`** that implements the ***specific decision tree*** above. Your program will read lines of data from a text file, print out the attribute values from that line, execute the decision tree process, and print the results.

Each line in the input file will have the following format:

```
soil_type,slope_in_percent,rainfall_in_mm,elevation_in_m,irrigated
```

For instance:

```
loam,12,350,300,true
clay,17.3,35,1200,false
gravel,24.5,2000,20,false
```

Note that the slope is floating point while the rainfall and elevation are integers. Note also that the soil type might be something other than the types *loam*, *clay* or *sandy.* If the soil type is not one your program knows about, you should print a message and skip this line.

To receive the best score on this assessment, you should write your program using the following guidelines:

1. Do not create a deeply nested set of `if...else if...else` statements to try and represent the entire tree. This is hard to understand, hard to modify and prone to errors. Instead, figure out a way to make the logic a) clear and b) easy to modify if we wanted to change some of the detailed tests in the decision tree.
2. Do not use a series of if statements that concatenate all the tests into a single compound condition. This has the same problems as the first approach.

Your program should accept the name of the input file on the command line, as a command line argument. It should be able to handle ***any input file*** that has the format above. Be sure to check that your attempt to open the input file is successful. If any of the lines in the input file have fewer than five data items, print an error message and skip that line.

Here is a sample run.

```
./cropDecision landInfo1.txt
Line 1:
    Soil type: loam
    Slope: 2.70%
    Rainfall: 10 mm
    Elevation: 1000 m
    Irrigated: yes
    Recommended crop: Rice

Line 2:
    Soil type: sandy
    Slope: 12.50%
    Rainfall: 200 mm
    Elevation: 5 m
    Irrigated: no
    Recommended crop: N/A

Line 3:
    Soil type: clay
    Slope: 15.00%
    Rainfall: 300 mm
    Elevation: 3200 m
    Irrigated: yes
    Recommended crop: N/A

Line 4:
    Soil type: loam
    Slope: 22.00%
```

```
    Rainfall: 400 mm
    Elevation: 5 m
    Irrigated: no
    Recommended crop: Tea

Line 5:
    Soil type: loam
    Slope: 22.00%
    Rainfall: 400 mm
    Elevation: 5 m
    Irrigated: yes
    Recommended crop: Tea

Line 6:
    Soil type: humus
    Slope: 10.00%
    Rainfall: 10 mm
    Elevation: 120 m
    Irrigated: no
    Unknown soil type - skipping


Line 7:
    Only 3 fields found - skipping

Line 8:
    Only 1 fields found - skipping

Line 9:
    Soil type: sandy
    Slope: 30.00%
    Rainfall: 200 mm
    Elevation: 100 m
    Irrigated: yes
    Recommended crop: Tea

Line 10:
    Soil type: clay
    Slope: 0.00%
    Rainfall: 30 mm
    Elevation: 4 m
    Irrigated: yes
    Recommended crop: Vegetables
```