121COM: Introduction to Computing

Academic Year 2015/16

LabSheet 2

For use in labs the week beginning Mon 5th October 2015

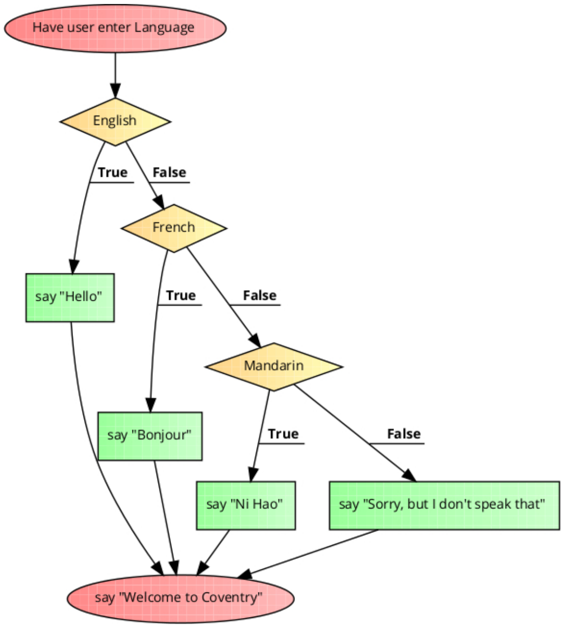
Author: Dr Matthew England (Coventry University)

C Lab Exercises 1 - Modelling Control

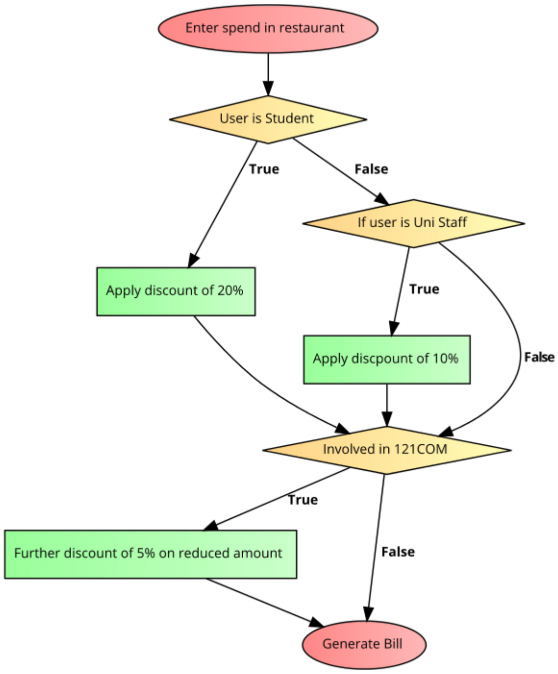
Q1 Define two variables `hours` and `minutes` which store the time in 24 hour clock notation. From this have Python print a sentence telling the time in 12 hour clock notation. For example, if the time is 15:25 Python should print `The time is 3:25pm`.

Q2 Download the file `Week2BuggyCode.py` from Moodle. Fix the multiple errors within so that it runs as described in the comments.

Q3 Write Python scripts which implement the following flow charts using conditionals. Note what type of conditionals you are using (chained, nested, separate).



(a)



(b)

Q4 Draw flow diagrams which represent the 4 scripts you implemented in Worksheet 2 Exercises Q2.

I Lab Exercises 2 - Taking Control

Q5 Extend the solution to Q1 above:

(a) Allow the user to give the initial hours and minutes. But check they are giving input in the correct form (printing your own error message otherwise).

(b) Have Python instead print the approximate time correct to the nearest 15 minutes. For example, if the time is 15:25 Python should print `The time is about half past three in the afternoon`.

Q6 In the first two worksheets we have met: numerical operators (`+`, `-`, `*`, `/`, `%`, `//`); comparison operators (`<`, `<=`, `>`, `>=`, `==`, `!=`); and logic operators (`not`, `or`, `and`). Write down the exact order of precedence for these operators (the order Python evaluates them in). Remember that Python always evaluates things in parentheses first. You can use this fact to create examples which demonstrate the order. E.g.

```
>>> (1+2)*3
9
>>> 1+(2*3)
7
>>> 1+2*3
7
```

This shows that Python performs multiplication before addition.

Q7 Download (i.e. Pull) the following repository from GitHub:
`https://github.com/Matthew-England/121COMWeek2Ex7`
It contains two files: a Python script **TextAdventure.py** which has code starting a text adventure game; and a text file `ReadMe.txt` setting out how the game works.

Pick one of the branches of the if-statement and continue the story. Add new options for the user to choose and new outcomes. Hint: Have the different branches of the story converge in places to avoid too many nested conditionals. After each addition test the code still works then version control in your local repository with Git.

A **Extension:** Find another student in the class who extended the code in the other branch. Hint: Agree this with your neighbour first. Merge your text adventure with them via GitHub to make a single larger adventure.

A Extended Task - Music APIs

Download (i.e. Pull) the following repository from GitHub:
`https://github.com/Matthew-England/121COMWeek2ExTask.git`
This contains two files I have written:

- `MusicAPIFunctions.py` defines functions for obtaining information related to music from the internet;
- `LinkingMusicAPIs.py` gives examples of how to use them.

Focus on the second file (you will understand everything in the first as we progress through the module).

1. Build a script for linking the services. Have Python ask the user questions so that they can use the services without knowing the syntax of my functions.
2. Allow the user to provide pairs of artists to see which is more popular (according to Spotify).

Work in pairs, with one on each task. Version control in Git and merge via GitHub.