Cashing Up Application Development Guide

We are looking to develop an application to be used to assist people in cashing up tills. In developing this application, we also aim to teach Nathan the basics of programming, and the python language, and eventually teach the both of us some more advanced aspects, including GUI development and product deployment.

This document aims to outline the requirements of the application in order to guide production, as well as to try and structure the development in a way which will logically introduce aspects of programming to Nathan in an applied way.

APPLICATION OUTLINE

The application will need to:

* Calculate the value of the contents of the till
  + –method tbd (currently expecting to ask the user to input the amount of coins of each value) –
* Minus the starting float from the total value of the til contents to give the ‘Takings’
* Generate a report (preferably printable) of the days takings.
* …

It is expected that there will be 2 versions of the applications.

The first will implement the actual functionality of the application. (collecting user input, calculating values etc…) and will run in the terminal. This is because any GUI aspects are beyond the scope of Nathans current understanding of programming, and I will need to research the possibilities of GUI modules with python.

The second implementation will then look to build up a ‘complete’ application, hooking up v1 functionality to a GUI, and add features such at the report generation, and eventually look to ‘deploy’ the application for use in a ‘live’ environment. As I am unsure of the requirements for most of this version, it is currently difficult to propose a timeline. Once development has begun, I will update timing expectations.

Technical specifications:

* The application will be developed with python
* The dev environment will be Python 3.6
* In order to give Nathan the opportunity to learn and explore, there will not be a ‘combined’ development project. We will both work on separate applications.
* Git will be used to share work for whatever reason.
* For version 2
  + Looking to deploy on all platforms, but atleast to windows.
  + GUI library TBC
  + …
* …

GUI DESIGN TO BE DISCUSSED AFTER VERSION 1 IS COMPLETE

TEACHING SCHEDULE:

Variables:

Variables are ways of ‘storing’ values that may or may not change during development. The terminology is you ‘declare’ a variable, and then assign to it.

Explain how in most cases, this also tells the computer to reserve a block big enough for that value type

Pyhton is nice that it automatically handles all of this. IT know if it’s a string or int or whatever and handles it all for you.

Aswell as the ‘basic’ variable types, you can also declare containers.

Containers are objects that hold many variables.

So for example, we could declare car1, car2, car3 etc…

Or we could declare a container called cars (which would be a list in python). From here we can declare this list as containing many cars ( =[car1, car2])

We can add things and remove things from a list, using append or take things out with remove (there is a lot more we can do but this is the basic)  
We can ‘access’ any item in a list with its index.

Python, like in mnay languages indexes from 0, so the first item is numbered ‘0’. So for our example, cars[0] would give you car1

Now… with all of these variables, they are ‘objects’, and because of this we can access their operators, which are basic built in features. So we can add them etc…. don’t worry too much about this. I can expand if needed.

While we’re here, we can assign anything to a variable, so lets try assigning user input to a variable.

Conditionals:

Now, that we have our variables, we may want to do different things depending on them. Without conditionals or loops, they same things would happen every time we ran our program, (the only difference might be the variables)

To do more interesting things we need to use conditionals. These are really simple to understand in python, as they tend to be the same as you might say them in plain English.

Conditionals used logic, at the very lowest level we check for true or false. The basic conditional is the if statement. Here we can ask the program a question and if the answer is true, it will do the code ‘inside’ the if statement, otherwise it will skip it.

We can extend this with the else. If the initial question returns true, it will do the code inside the ‘if’ block, other wise, if its false, we do the code in the else block.