Coffee Barista Assistant

Overview

Ordering hot drinks, such as coffee, at a cafe can be a familiar experience to most students. This assignment takes advantage of students' background knowledge of ordering at a cafe, so they can focus on the programming assignment's learning objectives. The assignment has different challenge layers to encourage the student to attempt increasingly difficult levels of the assignment, while still making the assignment fun and engaging.

Meta Information

Meta Name	Description
Summary	The Coffee Barista Assistant has students develop a tool that generates instructions to build a customer's cup of coffee. The assignment has a basic path for all students to complete, while containing additonal layers to challenge high performing students. Prerequisites for the assignment: • variables • integers • if then/else • loops • functions • lists
Audience	The assignment is appropriate for CS1 students as an assessment presented in the middle of the course, where lists have already been addressed.
Difficulty	The presented problem has different challenge layers that a teacher can decide to present. The teacher has the option to remove those layers and present a single basic workflow for students to solve. The assessment presented is relatively easy.
Topics	Lists, variables, and I/O concepts. The assignment also has the possibility of visualisation, depending on the course programming language. The visualisation component could have students graphically represent the coffee ordered.
Strengths	The assignment has multiple layers that a teacher can decide to present in one assessment, or use the layers over several assessments, building on the context so that students can focus on the learning objectives.
Weakness	If working with visual components, students might spend an excessive amount of time making the prepared coffee look realistic. The teacher has the choice to provide constraints on this requirement, so that students might devote more time to other tasks in the assignment.
Dependencies	Requires the student to have some basic knowledge of ingredients to make a coffee, and understand I/O to take the order. The program requires input from a user via the keyboard.
Variants	There are a few ways to present this problem: 1. Provide a Barista the ingredients to make a customer a cup of coffee. 2. Generate coffee instructions from pre-existing rules with additional customisations defined by the customer. 3. Generate a visual cup of coffee after all the correct information is provided by the user.

Pedagogy

This assignment can serve as a pre-activity to help students think about how they would design their programming solution. The teacher can get students to perform an unplugged design activity that describes the workflow for taking the customer's coffee order. By having students perform the unplugged design activity, they might be able to observe the question asking pattern that is encapsulated in the function. The teacher can ask students to use the results from the pre-activity as their guide for implementing a programming solution.

The assignment is presented using a Multi-layer Educational Design Pattern [1]. The pattern incorporates 'a variety of educational objectives into a single assignment by including the concepts on multiple knowledge and process levels' [1].

Extending the Assignment

The teacher can extend the assignment by incorporating visual components from the programming language, to visualise the oridered coffee. The assignment can be interactive by having coffee ingredients altered in real-time, affecting the visual representation of the ordered coffee. The teacher can decide to build on the problem with another assignment, by including other drinks with different requirements. The first assignment can be adjusted to include visualisation and interactivity by:

- Replacing the milk question with an interactive sliding scale, to depict the amount of milk to include in the coffee.
- Changing the coffee container when the customer asks for the coffee to go.
- Including an image of sugar when the student clicks on a designated part of the screen.

By continuing the problem as another assignment, the student gets the opportunity to practice code refactoring. The activity can be extended to include a checkForBadInput() function that encapsulates the logic for the data validation. Adding this function can also serve as a refactoring activity on the original solution.

Resources

Here are some resources to help with adopting this assignment.

- Assignment Description - An example handout of the program description.
- Solution - A coded solution to the problem written in Python.
- Examples - Sample execution, that gives teachers some use cases for testing the program.

Reference

[1] Christian Köppe, Leo Pruijt. Improving Students' Learning in Software Engineering Education through Multi-Level Assignments. *CSERC '14 Proceedings of the Computer Science Education Research Conference* (Berlin, Germany), 57-62, November 2014.