



PROGRESS OUTCOME 7

# Ordering smoothies

## Context

Jordan's class has been learning how to develop a computer program that uses a GUI (graphical user interface) and responds to users' inputs from button clicks, text input, or selections from drop-down menus or radio buttons.

The students have been given a scenario of developing an app for ordering smoothies from the school canteen. They have had to think about the design of the GUI and the program, and the testing needed to ensure the program functions accurately.



## Insight 1: Designing the GUI

We asked the school canteen about the process when students and teachers order smoothies. They gave us the following information about smoothie choices:

- three dairy choices (customers can only pick one)
- three size choices (customers can only pick one)
- six mixture choices (customers can only pick one).

I decided that the GUI would need three radio buttons for the dairy choices, three radio buttons for size choices and six radio buttons for the mixture. There would also need to be a button for clearing the order, a button to submit the order and a text box to display the order on the screen. I sketched out the layout and realised that there was no place to enter a name for the order, so I added another text box for input.



## Insight 2: Decisions about variables and collections

Looking at my design and thinking about how the program should function, I decided that I would need integer variables to store the choices for dairy, size, and mixture, and a string variable to store the name. I decided to store the different options for each choice in arrays so it would be easy to build the order. I could have built the order by hard-coding the output with a series of switch statements, but I thought this would take longer to code, which could introduce more errors and would be harder to update if the choices changed. Using arrays, I could loop through them to find the choices and build the order.



### Insight 3: Documenting and testing

After I had used a form design tool to build my GUI and initialised my variables and collections, I decided to program and test each event to make sure it worked as I expected. As each event should work in a similar way, I thought it would be more efficient to have one functioning correctly before programming the rest. I made sure all my variable names were clear and used the camelCase convention we had been taught in class. I then wrote some pseudocode for the first radio button click event and used that for my code comments.

I began testing by selecting a dairy choice and checking whether the correct choice displayed in the order output on the screen. At first this didn't work as I expected, but then I remembered that arrays are 0-indexed and I had set my choices to start with 1. Finding this out early on enabled me to get the codes to work correctly for the other radio button click events.



### Insight 4: Testing and debugging the order output

My order output had the correct information, but it wasn't readable because all the choices were joined together, such as in "Charlietrimlargebananarama". I had used string concatenation to build the output, but I had forgotten to add in carriage returns and newline commands. I added these commands and tested that the output was easier to read.