



3. WRITTEN RESPONSES (CREATED INDEPENDENTLY)

Submit your responses to prompts 3a – 3d, which are described below. Your response to all prompts combined must not exceed 750 words (program code is not included in the word count). Collaboration is **not** allowed on the written responses. Instructions for submitting your written responses are available on the [AP Computer Science Principles Exam Page](#) on AP Central.

3 a. Provide a written response that does all three of the following:

Approx. 150 words (for all subparts of 3a combined)

i. Describes the overall purpose of the program

The purpose of the program is to help coffee drinkers and café customers know more about what they're ordering when they are faced with all of the fancy coffee names on menus so that they can make more informed decisions about what to order.

ii. Describes what functionality of the program is demonstrated in the video

The program allows users to choose different coffee types from a dropdown menu to see a visual representation of what that type of coffee contains and at what proportions.

iii. Describes the input and output of the program demonstrated in the video

The input in the program consists of a user selection from a dropdown menu and the output is a visual representation of what that type of coffee contains and at what proportions.

- 3b.** Capture and paste two program code segments you developed during the administration of this task that contain a list (or other collection type) being used to manage complexity in your program.

Approx. 200 words (for all subparts of 3b combined, exclusive of program code)

- i. The first program code segment must show how data have been stored in the list.

```
var ingredientColours = {"Espresso": "#54443d", "Milk": "#f0f0f0",  
"Steamed Milk": "#f0f0f0", "Foam": "#ebe5bc", "Hot Chocolate":  
"#4a4543", "Water": "#a8dcff"};
```

- ii. The second program code segment must show the data in the same list being used, such as creating new data from the existing data or accessing multiple elements in the list, as part of fulfilling the program's purpose.

```
setFillColor(ingredientColours[ingredient]);  
setStrokeColor(ingredientColours[ingredient]);  
rect(5, yPos, 110, proportion * 30);
```

Then, provide a written response that does all three of the following:

- iii. Identifies the name of the list being used in this response

```
ingredientColours
```

- iv. Describes what the data contained in the list represent in your program

The data contained in this list consists of key-value pairs that represent coffee types and hex colour codes that correspond to those coffee types.

- v. Explains how the selected list manages complexity in your program code by explaining why your program code could not be written, or how it would be written differently, if you did not use the list

The list manages complexity by gathering all colour codes into one place and connects them immediately to their corresponding coffee types. If this was not the case, I would need to store the colour codes separately in individual variables and even then figuring out which value corresponds to which coffee type would not be possible.

DEFINITION:

List

A **list** is an ordered sequence of elements. The use of lists allows multiple related items to be represented using a single variable. Lists may be referred to by different names, such as **arrays**, depending on the programming language.

DEFINITION:

Collection Type

A **collection type** is a type that aggregates elements in a single structure. Some examples include lists, databases, hash tables, dictionaries, and sets.

IMPORTANT:

The data abstraction must make the program easier to develop (alternatives would be more complex) or easier to maintain (future changes to the size of the list would otherwise require significant modifications to the program code).

- 3 c.** Capture and paste two program code segments you developed during the administration of this task that contain a student-developed procedure that implements an algorithm used in your program and a call to that procedure.

Approx. 200 words (for all subparts of 3c combined, exclusive of program code)

- i. The first program code segment must be a student-developed procedure that:
- ☐ Defines the procedure's name and return type (if necessary)
 - ☐ Contains and uses one or more parameters that have an effect on the functionality of the procedure
 - ☐ Implements an algorithm that includes sequencing, selection, and iteration

```
function setCanvas(coffeeType) {
  initCupImg();
  var ratios = coffeeRatios[coffeeType];
  var yPos = 197;
  clearIngredientLabels();
  for (var i = 0; i < Object.keys(ratios).length; i++) {
    var ingredient = Object.keys(ratios)[i];
    var proportion = ratios[ingredient];
    yPos = yPos - (proportion * 30);
    setStrokeWidth(1);
    setFillColor(ingredientColours[ingredient]);
    setStrokeColor(ingredientColours[ingredient]);
    rect(5, yPos, 110, proportion * 30);
    setText("label"+parseInt((200-yPos+3) / 30), ingredient);
  }
}
```

- ii. The second program code segment must show where your student-developed procedure is being called in your program.

```
onEvent("coffeeSelector", "change", function( ) {
  setCanvas(getText("coffeeSelector"));
});
```

Then, provide a written response that does both of the following:

- iii. Describes in general what the identified procedure does and how it contributes to the overall functionality of the program

This function is what creates and updates the visual representation of the selected type of coffee. This function is automatically run when the user selects a new type of coffee thereby immediately matching the visuals on screen to the user selection.

- iv. Explains in detailed steps how the algorithm implemented in the identified procedure works. Your explanation must be detailed enough for someone else to recreate it.

This function starts by calling the `initCupImg()` function which clears the current image from the screen and replaces it with the image of an empty cup. It then gets the ratio of ingredients of the coffee type specified in the parameter from the `coffeeRatios` object, sets a variable for the y-coordinate to 197 (3 less than the height of the image canvas to account for the line width of the cup), and calls the `clearIngredientLabels()` function to remove all previous text from the ingredient labels in the visual representation. It uses a for-loop that runs a number of times equivalent to the number of ingredients in the specified coffee type. Within the for-loop, the ingredient matching the iteration count is accessed along with its proportion and both values are stored in variables. The y-coordinate is reset to 30 times the proportion of the ingredient less than its previous value to account for the amount of space that proportion will use in the canvas. The stroke width is then set to one, the fill colour and the stroke colour are set to the ingredients corresponding colour from the `ingredientColours` object, and a rectangle is drawn from the y-coordinate and an x-index of 5 at a width of 110 and a height of `proportion * 30`. Finally, the corresponding ingredient label's text is set to the ingredient name.

IMPORTANT:

Built-in or existing procedures and language structures, such as event handlers and main methods, are not considered student-developed.

3d. Provide a written response that does all three of the following:

Approx. 200 words (for all subparts of 3d combined)

- i. Describes two calls to the procedure identified in written response 3c. Each call must pass a different argument(s) that causes a different segment of code in the algorithm to execute.

First call:

```
setCanvas("Americano");
```

Second call:

```
setCanvas("Macchiato");
```

- ii. Describes what condition(s) is being tested by each call to the procedure

Condition(s) tested by the first call:

The conditions tested by this call are the name of the coffee drink (Americano) which provides access to the names of its ingredients (espresso and water), and subsequently the proportions (3 and 3) of each ingredient and the hex codes for the colours representing those ingredients on the canvas (#54443d and #a8dcff). This information is then used to generate a visual representation that is specific to the "Americano" option in the dropdown menu.

Condition(s) tested by the second call:

The conditions tested by this call are the name of the coffee drink (Macchiato) which provides access to the names of its ingredients (espresso and milk), and subsequently the proportions (4 and 2) of each ingredient and the hex codes for the colours representing those ingredients on the canvas (#54443d and #f0f0f0).

- iii. Identifies the result of each call

Result of the first call:

This information is used to generate a visual representation that is specific to the "Americano" option in the dropdown menu which displays an image of a cup containing two rectangles at a proportion of 1:1 coloured to represent the ingredients listed on the left of the cup, espresso and water.

Result of the second call:

This information is used to generate a visual representation that is specific to the "Macchiato" option in the dropdown menu which displays an image of a cup containing two rectangles at a proportion of 4:2 coloured to represent the ingredients listed on the left of the cup, espresso and milk.