PREPARED BY DAUD EID

AUGUST 11, 2025, 9:10PM

Q6

Performance Report

MINI CAFÉ ORDERING SYSTEM (PYTHON CODE)

ARMYMAN10249.WIXSITE.COM/MY-SITE-4 252-63-3227827 DAUDSHAFAD@YAHOO.COM

Alright — here's your **full written evaluation** of your café program, broken down step-by-step with strengths, issues, and suggestions.

Code Evaluation Report – Café Ordering Program

1. Variable Definitions

cafe_name = "joy cafe"

items_dict = {"cappucino" : 5.3, "tea": 2.4, "coffee" : 6.8}

✅ **Strengths:**

- Clear naming for cafe_name.
- Dictionary is correctly defined with keys as strings and prices as floats.

⚠ **Issues:**

- "cappucino" is misspelled — correct spelling is "cappuccino".
- Dictionary keys are case-sensitive, so if the user types "Tea" instead of "tea", it won't match.

💡 **Suggestion:**

- Use .lower() when checking user input to avoid case sensitivity issues.

---

2. Customer Name Input

customer_name = str(input("Enter your name please: "))

✅ **Strengths:**

- Correctly takes string input.
- str() conversion is unnecessary because input() already returns a string.

⚠ **Issues:**

- Could be improved by stripping spaces: input("...").strip().

---

3. Converting Dictionary Keys to Tuple

items_tuple = tuple(items_dict)

print(items_tuple)

✅ **Strengths:**

- Correctly converts dictionary keys to tuple and displays available items.

⚠ **Issues:**

- Printing them directly shows them in parentheses form — could be formatted for clarity.

💡 **Suggestion:**

```
print("Available items:", ", ".join(items_tuple))
```

---

4. Customer Choice and Quantity

```
customer_choice = str(input("Enter the item you want to buy by typing it's name: "))
number_of_items = int(input("How many of this item you want: "))
```

✅ **Strengths:**

- Uses int() for quantity.
- Clear prompts for user input.

⚠ **Issues:**

- "it's" is grammatically incorrect — should be "its".
- No input validation for quantity (e.g., negative numbers).

---

5. Item Check and Price Calculation

```
if customer_choice in items_dict:
  customer_choice_price = items_dict[customer_choice]
  total_cost = customer_choice_price * number_of_items
  print(total_cost)
else:
  print("Item not found, error.")
```

✅ **Strengths:**

- Correct dictionary access and multiplication.
- Good use of if for validation.

⚠ **Issues:**

- Again, case sensitivity may cause mismatches.
- If else runs, total_cost will not exist but is still printed later — will cause an error.

💡 **Suggestion:**

- Initialize total_cost = 0 before the if block.

---

6. Exit Statement

```
exit
```

⚠ **Issues:**

- This does nothing because exit must be a function call: exit().

💡 **Suggestion:**

- If you want to stop the program here, use:

```
import sys
sys.exit()
```

- Or just let the program continue without it.

---

7. Match Statement for Eating Option

```
customer_situation = str(input("Do you want to (1. eat in) or (2.take away) by typing it: "))
match customer_situation:
  case "eat in":
    print("Welcome, feel relax to your choice.")
  case "take away":
    print("Thanks, have a peace out.")
  case _ :
    print("Invalid option!")
```

✅ **Strengths:**

- Correct usage of match statement.
- Good handling of invalid inputs.

⚠ **Issues:**

- In the prompt, you say (1. eat in) but check for "eat in", not "1". This could confuse the user.
- Spelling/grammar issues: "feel relax" → "feel relaxed".

---

8. Final Confirmation Message

```
print("\n---------Confirmation message---------")
print(f"Welcom to {cafe_name.upper()}.")
print(f"Your name is: {customer_name.upper()}.")
print(f"You ordered a {customer_choice.upper()}")
print(f"You want {number_of_items} of this item.")
print(f"Your Total cost is: ${total_cost}.")
print(f"{customer_situation.upper()}.")
```

✅ **Strengths:**

- Good use of f-strings.

- Nice uppercase formatting for emphasis.

⚠ **Issues:**

- "Welcom" → "Welcome".
- If the item wasn't found earlier, total_cost will cause an error here.

Final Evaluation

Overall Functionality:
The program mostly works as intended for a simple café order system. However, it has **case sensitivity issues**, **a potential runtime error with total_cost**, **a non-functional exit statement**, and **a mismatch between prompt and match statement values**.

Score: 80% ✅

Reason: Good understanding of variables, tuples, dictionaries, if/else, match, and inputs. Needs improvements in error handling, spelling, and code robustness.

If you want, I can now rewrite your code **fixed and improved** but still **without functions or for loops**, keeping it within your allowed topics. This would make it work smoothly without errors. Would you like me to do that?