**ISYS2001** **Extended Learning Portfolio Ai-reference sheet**

Oliver Jamieson - 21443945

**Question 1**

Please write a simple code in python that asks for the user's monthly income, as well the cost of their expense categories (rent, groceries, utilities and entertainment)

Sure, here's a simple Python code that does what you've described:

python

Copy code

def get\_user\_input(prompt):

while True:

user\_input = input(prompt)

if user\_input.isdigit():

return int(user\_input)

else:

print("Please enter a valid number.")

def main():

print("Welcome to the monthly budget calculator!")

# Get user's monthly income

monthly\_income = get\_user\_input("Enter your monthly income: $")

# Get user's monthly expenses

rent = get\_user\_input("Enter your monthly rent expense: $")

groceries = get\_user\_input("Enter your monthly groceries expense: $")

utilities = get\_user\_input("Enter your monthly utilities expense: $")

entertainment = get\_user\_input("Enter your monthly entertainment expense: $")

# Calculate total expenses

total\_expenses = rent + groceries + utilities + entertainment

# Calculate remaining income after expenses

remaining\_income = monthly\_income - total\_expenses

# Display the budget summary

print("\n--- Budget Summary ---")

print(f"Monthly Income: ${monthly\_income}")

print(f"Total Expenses: ${total\_expenses}")

print(f"Remaining Income: ${remaining\_income}")

if \_\_name\_\_ == "\_\_main\_\_":

main()

**Question 2**

How would i add error handling: def schedule\_volunteers(volunteers: list, shifts: list) -> dict: #explain what this is """ Assigns volunteers to available shifts based on their preferences. >>> schedule\_volunteers(["Alice", "Bob"], ["Morning", "Afternoon"]) {'Morning': 'Alice', 'Afternoon': 'Bob'} """ # Initialize an empty dictionary for assignments assignments = [] #change the name for v in volunteers: # Loop through the volunteers # Assuming user has entered a valid shift number # Get the shift preference # User selects preferred shift number within range preference\_num = int(input(f"{v}, enter your preferred shift number (1-{len(shifts)}): ")) preference\_num -= 1 # Convert to 0-based index # Assigns preferred shift to worker, updates assignments, and returns assignments.append(shifts.pop(preference\_num), v) return assignments if \_\_name\_\_ == "\_\_main\_\_": # List of volunteers vols = ["Alice", "Bob"] # List of shifts shifts = ["Morning", "Afternoon"] # Call the schedule\_volunteers function result = schedule\_volunteers(vols, shifts) # Print the results print(result)

ChatGPT

ChatGPT

To add error handling to the code, particularly for cases where the user enters an invalid shift number, you can use a try-except block. Here's how you can modify the code to include error handling:

python

Copy code

def schedule\_volunteers(volunteers: list, shifts: list) -> dict:

"""

Assigns volunteers to available shifts based on their preferences.

:param volunteers: A list of volunteers.

:param shifts: A list of shifts.

:return: A dictionary mapping shifts to assigned volunteers.

>>> schedule\_volunteers(["Alice", "Bob"], ["Morning", "Afternoon"])

{'Morning': 'Alice', 'Afternoon': 'Bob'}

"""

# Initialize an empty dictionary for shift assignments

shift\_assignments = {}

for volunteer in volunteers:

while True:

try:

# Get the shift preference from the volunteer

preference\_num = int(input(f"{volunteer}, enter your preferred shift number (1-{len(shifts)}): "))

if 1 <= preference\_num <= len(shifts):

break

else:

print("Please enter a valid shift number.")

except ValueError:

print("Please enter a valid number.")

# Assign the preferred shift to the volunteer

preference\_index = preference\_num - 1 # Convert to 0-based index

shift\_assignments[shifts[preference\_index]] = volunteer

return shift\_assignments

if \_\_name\_\_ == "\_\_main\_\_":

# List of volunteers

volunteers = ["Alice", "Bob"]

# List of shifts

shifts = ["Morning", "Afternoon"]

# Call the schedule\_volunteers function

assignments = schedule\_volunteers(volunteers, shifts)

# Print the results

print(assignments)

In this modified version, the try-except block is used to catch any ValueError that occurs when trying to convert the user input to an integer. If a ValueError is raised (i.e., if the input cannot be converted to an integer), it prints a message asking the user to enter a valid number. This ensures that the program doesn't crash if the user enters invalid input.

**Question 3**

Write a simple python code that will allow someone to order at a cafe, using conditionals, with the menu options being a burger, salad or pasta, each meal coming with a side option of chips, a coke or no side

ChatGPT

ChatGPT

Here's a simple Python code that allows someone to order at a cafe with menu options for a main meal (burger, salad, or pasta) and a side option (chips, coke, or no side):

python

Copy code

def order\_at\_cafe():

print("Welcome to the cafe!")

print("Here's our menu:")

print("1. Burger")

print("2. Salad")

print("3. Pasta")

# Get user's main meal choice

main\_meal = input("Please enter the number corresponding to your main meal choice: ")

# Determine main meal choice

if main\_meal == "1":

main\_meal\_choice = "Burger"

elif main\_meal == "2":

main\_meal\_choice = "Salad"

elif main\_meal == "3":

main\_meal\_choice = "Pasta"

else:

print("Invalid choice. Please select a valid option.")

return

# Offer side options

print("Great choice! Now, let's select a side option:")

print("1. Chips")

print("2. Coke")

print("3. No side")

# Get user's side choice

side\_option = input("Please enter the number corresponding to your side option: ")

# Determine side choice

if side\_option == "1":

side\_choice = "Chips"

elif side\_option == "2":

side\_choice = "Coke"

elif side\_option == "3":

side\_choice = "No side"

else:

print("Invalid choice. Please select a valid option.")

return

# Display order summary

print("Your order summary:")

print(f"- Main meal: {main\_meal\_choice}")

print(f"- Side option: {side\_choice}")

print("Thank you for your order! Enjoy your meal!")

# Call the function to take the order

order\_at\_cafe()

This code presents the user with a menu of main meal options (burger, salad, or pasta) and then asks the user to select a main meal and a side option. It then displays the order summary including the main meal and side option chosen by the user.

You

Add code so that after the order is placed, the customer will be told of the price. burger's are $15, salad's are $12, and Pasta's are $17.50. Fries and coke both cost $5, no side is free

ChatGPT

ChatGPT

Here's the updated code with the prices added:

python

Copy code

def order\_at\_cafe():

print("Welcome to the cafe!")

print("Here's our menu:")

print("1. Burger - $15")

print("2. Salad - $12")

print("3. Pasta - $17.50")

# Get user's main meal choice

main\_meal = input("Please enter the number corresponding to your main meal choice: ")

# Determine main meal choice and its price

if main\_meal == "1":

main\_meal\_choice = "Burger"

main\_meal\_price = 15

elif main\_meal == "2":

main\_meal\_choice = "Salad"

main\_meal\_price = 12

elif main\_meal == "3":

main\_meal\_choice = "Pasta"

main\_meal\_price = 17.50

else:

print("Invalid choice. Please select a valid option.")

return

# Offer side options

print("Great choice! Now, let's select a side option:")

print("1. Chips - $5")

print("2. Coke - $5")

print("3. No side - Free")

# Get user's side choice

side\_option = input("Please enter the number corresponding to your side option: ")

# Determine side choice and its price

if side\_option == "1":

side\_choice = "Chips"

side\_price = 5

elif side\_option == "2":

side\_choice = "Coke"

side\_price = 5

elif side\_option == "3":

side\_choice = "No side"

side\_price = 0

else:

print("Invalid choice. Please select a valid option.")

return

# Calculate total price

total\_price = main\_meal\_price + side\_price

# Display order summary including the total price

print("Your order summary:")

print(f"- Main meal: {main\_meal\_choice} (${main\_meal\_price})")

print(f"- Side option: {side\_choice} (${side\_price})")

print(f"Total price: ${total\_price}")

print("Thank you for your order! Enjoy your meal!")

# Call the function to take the order

order\_at\_cafe()