Pre Lab 4

Pattern thinking: Logic in Layers

Applied Python Programming with AI and Raspberry Pi Interfaces

Instructor: Dr. Vikas Thammanna Gowda Semester: ABCD 20YX

Points: 20 Assign: *TBD* Due: *TBD*

Name: _____

Follow these steps to deepen your understanding of Python loops (`while` and `for`). You'll work with the provided "Movie Ticket Sales Calculator." Code and answer each question by editing, running, observing, and documenting the results.

1. Set up your environment

- Open your favorite Python editor or IDE (e.g., IDLE, VS Code, PyCharm).
- Open the file `movie_ticket_functions.py`.
- Run the base code.
- Execute `movie_ticket_functions.py` as-is.
- Verify that it prints the subtotal, tax, shipping, total cost, and whether the order is within your budget.

```python
# Pricing & fees (constants)
base_ticket_price = 12.00     # $ per ticket
service_fee       = 1.50      # flat service fee per order

# 1. calculate_subtotal: returns subtotal for given qty of tickets
def calculate_subtotal(qty):
    return base_ticket_price * qty

# 2. determine_age_discount: returns discount amount based on age
def determine_age_discount(subtotal, age_group):
    if age_group == 'child':    # under 12
        return subtotal * 0.50  # 50% off
    elif age_group == 'senior': # 65 and over
        return subtotal * 0.30  # 30% off
    else:
        return 0.00             # no discount

# 3. apply_weekday_discount: 2 off per ticket on weekdays (Mon-Thu)
def apply_weekday_discount(qty, day_of_week):
    weekdays = ['Monday', 'Tuesday', 'Wednesday', 'Thursday']
    return 2.00 * qty if day_of_week in weekdays else 0.00

# 4. calculate_total: returns (total, subtotal, age_disc, weekday_disc,
    service_fee)
def calculate_total(qty, age_group, day_of_week):
    subtotal      = calculate_subtotal(qty)
    age_discount  = determine_age_discount(subtotal, age_group)
    weekday_disc  = apply_weekday_discount(qty, day_of_week)
    net           = subtotal - age_discount - weekday_disc
    total         = net + service_fee
    return total, subtotal, age_discount, weekday_disc, service_fee

# 5. process_orders: iterates orders and prints summary
```

```
def process_orders(orders):
    for i, order in enumerate(orders, start=1):
        qty, age_group, day = order
        total, subtotal, age_disc, wd_disc, fee = calculate_total(qty,
           age_group, day)
        print(f"Order {i}: qty={qty}, age={age_group}, day={day} -> \
              subtotal=${subtotal:.2f}, age_disc=${age_disc:.2f}, \
              weekday_disc=${wd_disc:.2f}, fee=${fee:.2f}, total=${total:.2f}")

# Main program
if __name__ == "__main__":
    # orders: (quantity, age_group, day_of_week)
    orders = [
        (4, 'adult', 'Friday'),
        (2, 'child', 'Wednesday')
    ]
    process_orders(orders)
```

2. Record your observations (4 Points)

orders	subtotal	age_dics	weekday_disc	fee	total
(4, 'adult', 'Friday')					
(2, 'child', 'Wednesday')					
(3, 'senior', 'Monday')					
(1, 'adult', 'Tuesday')					
(5, 'child', 'Saturday')					

3. Answer the following questions (6 Points)

1. Identify one parameter and one argument in the call to `calculate_total`.

2. Which functions call other functions? For each, name the **caller** and **callee**.

3. What happens if you remove the return statement from `determine_age_discount`? Describe the resulting error or behavior.

4. Thinker and Tinker

1. What happens if you pass an unexpected `age_group` value (e.g. 'teen')? How could you guard against that? **(6 Points)**

2. Suppose you want to give a $1 off per ticket for "matinee" showings (before noon). How would you extend the code to handle a fourth parameter `show_time`? **(4 Points)**

3. How would you add a 10% surcharge for groups larger than 6 tickets? At what point in `calculate_total` should this logic go, and why?

4. Notice both discount functions return dollar amounts rather than rates. Could you refactor them to return rates instead, and then compute all discounts in a single place? What would the updated function signatures look like?

5. Submission Instructions

- Drop off your completed work in the file folder outside my office door (West Hall 100).
- Turn it in in class before start of lab.
- Scan your work into a PDF and upload it to LLM.
- If you upload an image to LLM, combine all pages into a single, high-resolution file that is clear and easy to read. (Failure to follow this instruction will result loss of points.))