WEEK 6

TASK 1:

# Constants

ONE\_DAY\_PRICES = {

    "Adult": 20.00,

    "Child": 12.00,

    "Senior": 16.00,

    "Family (Up to 2 adults/seniors and 3 children)": 60.00,

    "Group (6 or more people) - Price per person": 15.00

}

TWO\_DAY\_PRICES = {

    "Adult": 30.00,

    "Child": 18.00,

    "Senior": 24.00,

    "Family (Up to 2 adults/seniors and 3 children)": 90.00,

    "Group (6 or more people) - Price per person": 22.50

}

EXTRA\_ATTRACTIONS = {

    "Lion Feeding": 2.50,

    "Penguin Feeding": 2.00,

    "Evening Barbecue (Two-Day Tickets Only)": 5.00

}

# Function to display ticket options

def display\_ticket\_options():

    print("One-Day Ticket Options:")

    print("{:<50} {:<15} {:<15}".format("Ticket Type", "Cost for One Day", "Cost for Two Days"))

    for ticket, price in ONE\_DAY\_PRICES.items():

        print("{:<50} ${:<15.2f}".format(ticket, price))

    print("\nTwo-Day Ticket Options:")

    print("{:<50} {:<15} {:<15}".format("Ticket Type", "Cost for One Day", "Cost for Two Days"))

    for ticket, price in TWO\_DAY\_PRICES.items():

        print("{:<50} ${:<15.2f}".format(ticket, price))

    print("\nExtra Attractions:")

    print("{:<50} {:<15}".format("Extra Attraction", "Cost per Person"))

    for attraction, price in EXTRA\_ATTRACTIONS.items():

        print("{:<50} ${:<15.2f}".format(attraction, price))

    print("\nDays Available for Booking: Any valid day")

# Main program

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

    display\_ticket\_options()

**TASK 2:**

# Constants

ONE\_DAY\_PRICES = {

    "Adult": 20.00,

    "Child": 12.00,

    "Senior": 16.00,

    "Family (Up to 2 adults/seniors and 3 children)": 60.00,

    "Group (6 or more people) - Price per person": 15.00

}

TWO\_DAY\_PRICES = {

    "Adult": 30.00,

    "Child": 18.00,

    "Senior": 24.00,

    "Family (Up to 2 adults/seniors and 3 children)": 90.00,

    "Group (6 or more people) - Price per person": 22.50

}

EXTRA\_ATTRACTIONS = {

    "Lion Feeding": 2.50,

    "Penguin Feeding": 2.00,

    "Evening Barbecue (Two-Day Tickets Only)": 5.00

}

# Function to display ticket options

def display\_ticket\_options():

    print("One-Day Ticket Options:")

    print("{:<50} {:<15} {:<15}".format("Ticket Type", "Cost for One Day", "Cost for Two Days"))

    for ticket, price in ONE\_DAY\_PRICES.items():

        print("{:<50} ${:<15.2f}".format(ticket, price))

    print("\nTwo-Day Ticket Options:")

    print("{:<50} {:<15} {:<15}".format("Ticket Type", "Cost for One Day", "Cost for Two Days"))

    for ticket, price in TWO\_DAY\_PRICES.items():

        print("{:<50} ${:<15.2f}".format(ticket, price))

    print("\nExtra Attractions:")

    print("{:<50} {:<15}".format("Extra Attraction", "Cost per Person"))

    for attraction, price in EXTRA\_ATTRACTIONS.items():

        print("{:<50} ${:<15.2f}".format(attraction, price))

    print("\nDays Available for Booking: Any valid day")

# Function to process a booking

def process\_booking():

    while True:

        total\_cost = 0

        booking\_number = generate\_booking\_number()

        # Input tickets required

        print("\nEnter the number of tickets required for each type:")

        for ticket, price in ONE\_DAY\_PRICES.items():

            num\_tickets = get\_valid\_input("Number of {} tickets: ".format(ticket), int)

            total\_cost += num\_tickets \* price

        # Input extra attractions required

        print("\nEnter the number of people for each extra attraction:")

        for attraction, price in EXTRA\_ATTRACTIONS.items():

            num\_people = get\_valid\_input("Number of people for {}: ".format(attraction), int)

            total\_cost += num\_people \* price

        # Display booking details

        print("\nBooking Details:")

        print("Booking Number:", booking\_number)

        print("Total Cost: ${:.2f}".format(total\_cost))

        # Ask if the user wants to make another booking

        choice = input("Do you want to make another booking? (Press any key to continue or 'exit' to quit): ")

        if choice.lower() == 'exit':

            break

# Function to generate a unique booking number

def generate\_booking\_number():

    # Simulate a unique booking number generation mechanism

    return "BK" + str(random.randint(1000, 9999))

# Function to get valid user input

def get\_valid\_input(prompt, data\_type):

    while True:

        try:

            user\_input = data\_type(input(prompt))

            if user\_input < 0:

                raise ValueError("Please enter a non-negative number.")

            return user\_input

        except ValueError as ve:

            print("Error:", ve)

# Main program for Task 2

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

    import random

    display\_ticket\_options()

    process\_booking()

**TASK 3:**

# Constants

ONE\_DAY\_PRICES = {

    "Adult": 20.00,

    "Child": 12.00,

    "Senior": 16.00,

    "Family (Up to 2 adults/seniors and 3 children)": 60.00,

    "Group (6 or more people) - Price per person": 15.00

}

TWO\_DAY\_PRICES = {

    "Adult": 30.00,

    "Child": 18.00,

    "Senior": 24.00,

    "Family (Up to 2 adults/seniors and 3 children)": 90.00,

    "Group (6 or more people) - Price per person": 22.50

}

EXTRA\_ATTRACTIONS = {

    "Lion Feeding": 2.50,

    "Penguin Feeding": 2.00,

    "Evening Barbecue (Two-Day Tickets Only)": 5.00

}

# Function to display ticket options

def display\_ticket\_options():

    print("One-Day Ticket Options:")

    print("{:<50} {:<15} {:<15}".format("Ticket Type", "Cost for One Day", "Cost for Two Days"))

    for ticket, price in ONE\_DAY\_PRICES.items():

        print("{:<50} ${:<15.2f}".format(ticket, price))

    print("\nTwo-Day Ticket Options:")

    print("{:<50} {:<15} {:<15}".format("Ticket Type", "Cost for One Day", "Cost for Two Days"))

    for ticket, price in TWO\_DAY\_PRICES.items():

        print("{:<50} ${:<15.2f}".format(ticket, price))

    print("\nExtra Attractions:")

    print("{:<50} {:<15}".format("Extra Attraction", "Cost per Person"))

    for attraction, price in EXTRA\_ATTRACTIONS.items():

        print("{:<50} ${:<15.2f}".format(attraction, price))

    print("\nDays Available for Booking: Any valid day")

# Function to process a booking

def process\_booking():

    while True:

        total\_cost = 0

        booking\_number = generate\_booking\_number()

        # Input tickets required

        print("\nEnter the number of tickets required for each type:")

        num\_adults = get\_valid\_input("Number of Adult tickets: ", int)

        num\_children = get\_valid\_input("Number of Child tickets: ", int)

        num\_seniors = get\_valid\_input("Number of Senior tickets: ", int)

        num\_families = get\_valid\_input("Number of Family tickets: ", int)

        num\_groups = get\_valid\_input("Number of Group tickets (6 or more people): ", int)

        total\_cost += (num\_adults \* ONE\_DAY\_PRICES["Adult"] +

                       num\_children \* ONE\_DAY\_PRICES["Child"] +

                       num\_seniors \* ONE\_DAY\_PRICES["Senior"] +

                       num\_families \* ONE\_DAY\_PRICES["Family (Up to 2 adults/seniors and 3 children)"] +

                       num\_groups \* ONE\_DAY\_PRICES["Group (6 or more people) - Price per person"])

        # Input extra attractions required

        print("\nEnter the number of people for each extra attraction:")

        num\_lion\_feeding = get\_valid\_input("Number of people for Lion Feeding: ", int)

        num\_penguin\_feeding = get\_valid\_input("Number of people for Penguin Feeding: ", int)

        num\_evening\_barbecue = get\_valid\_input("Number of people for Evening Barbecue: ", int)

        total\_cost += (num\_lion\_feeding \* EXTRA\_ATTRACTIONS["Lion Feeding"] +

                       num\_penguin\_feeding \* EXTRA\_ATTRACTIONS["Penguin Feeding"] +

                       num\_evening\_barbecue \* EXTRA\_ATTRACTIONS["Evening Barbecue (Two-Day Tickets Only)"])

        # Check if booking provides the best value

        is\_best\_value, savings = check\_best\_value(total\_cost, num\_adults, num\_children)

        # Display booking details

        print("\nBooking Details:")

        print("Booking Number:", booking\_number)

        print("Total Cost: ${:.2f}".format(total\_cost))

        if is\_best\_value:

            print("This booking is the best value.")

        else:

            print("It's better to buy two family tickets instead of a group ticket.")

            print("You could save ${:.2f} by choosing two family tickets.".format(savings))

        # Ask if the user wants to make another booking

        choice = input("Do you want to make another booking? (Press any key to continue or 'exit' to quit): ")

        if choice.lower() == 'exit':

            break

# Function to generate a unique booking number

def generate\_booking\_number():

    import random

    return "BK" + str(random.randint(1000, 9999))

# Function to get valid user input

def get\_valid\_input(prompt, data\_type):

    while True:

        try:

            user\_input = data\_type(input(prompt))

            if user\_input < 0:

                raise ValueError("Please enter a non-negative number.")

            return user\_input

        except ValueError as ve:

            print("Error:", ve)

# Function to check if a booking gives the best value

def check\_best\_value(total\_cost, num\_adults, num\_children):

    # Calculate the total cost of two family tickets

    family\_cost = ONE\_DAY\_PRICES["Family (Up to 2 adults/seniors and 3 children)"] \* 2

    # Calculate the total cost of a group ticket

    group\_cost\_per\_person = ONE\_DAY\_PRICES["Group (6 or more people) - Price per person"]

    group\_cost = group\_cost\_per\_person \* (num\_adults + num\_children)

    if group\_cost < family\_cost:

        return False, family\_cost - group\_cost  # Group ticket is more cost-effective

    else:

        return True, 0  # Two family tickets are more cost-effective

# Main program

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

    display\_ticket\_options()

    process\_booking()