**Assignment 5**

**Control Structure**

**SaiPrabath Chowdary S**

**Task 1: Conditional Statements**

In a BookingSystem, you have been given the task is to create a program to book tickets. if available tickets more than noOfTicket to book then display the remaining tickets or ticket unavailable:

**Tasks:**

1. Write a program that takes the availableTicket and noOfBookingTicket as input.
2. Use conditional statements (if-else) to determine if the ticket is available or not.

3. Display an appropriate message based on ticket availability.

def check\_ticket\_availability(available\_tickets, no\_of\_tickets\_to\_book):

if available\_tickets >= no\_of\_tickets\_to\_book:

remaining\_tickets = available\_tickets - no\_of\_tickets\_to\_book

print(f"Tickets available! Remaining tickets: {remaining\_tickets}")

else:

print("Tickets unavailable!")

# Example 1

available\_tickets = 50

no\_of\_tickets\_to\_book = 3

check\_ticket\_availability(available\_tickets, no\_of\_tickets\_to\_book)

Screenshot 2024-04-25 at 3.35.14 PM

# Example 2

available\_tickets = 5

no\_of\_tickets\_to\_book = 7

check\_ticket\_availability(available\_tickets, no\_of\_tickets\_to\_book)

Screenshot 2024-04-25 at 3.39.02 PM

**Task 2: Nested Conditional Statements**

Create a program that simulates a Ticket booking and calculating cost of tickets. Display tickets options such as "Silver", "Gold", "Dimond". Based on ticket category fix the base ticket price and get the user input for ticket type and no of tickets need and calculate the total cost of tickets booked.

def calculate\_ticket\_cost(ticket\_type, num\_tickets):

base\_prices = {"Silver": 100, "Gold": 200, "Diamond": 300}

if ticket\_type in base\_prices:

base\_price = base\_prices[ticket\_type]

total\_cost = base\_price \* num\_tickets

return total\_cost

else:

return None

ticket\_type = input("Enter ticket type (Silver/Gold/Diamond): ")

num\_tickets = int(input("Enter number of tickets: "))

total\_cost = calculate\_ticket\_cost(ticket\_type, num\_tickets)

if total\_cost is not None:

print(f"Total cost for {num\_tickets} {ticket\_type} tickets: {total\_cost}")

else:

print("Invalid ticket type!")



**Task 3: Looping**

From the above task book the tickets for repeatedly until user type "Exit"

def calculate\_ticket\_cost(ticket\_type, num\_tickets):

base\_prices = {"Silver": 100, "Gold": 200, "Diamond": 300}

if ticket\_type in base\_prices:

base\_price = base\_prices[ticket\_type]

total\_cost = base\_price \* num\_tickets

return total\_cost

else:

return None

while True:

ticket\_type = input("Enter ticket type (Silver/Gold/Diamond), or type 'Exit' to quit: ")

if ticket\_type.lower() == "exit":

print("Exiting ticket booking system.")

break

num\_tickets = int(input("Enter number of tickets: "))

total\_cost = calculate\_ticket\_cost(ticket\_type, num\_tickets)

if total\_cost is not None:

print(f"Total cost for {num\_tickets} {ticket\_type} tickets: {total\_cost}")

else:

print("Invalid ticket type!")

