***MINI Project Summary***

***On***

***Object Oriented Programming***

**Student Name**:/ **Roll No** :

Tushar Patil / A254

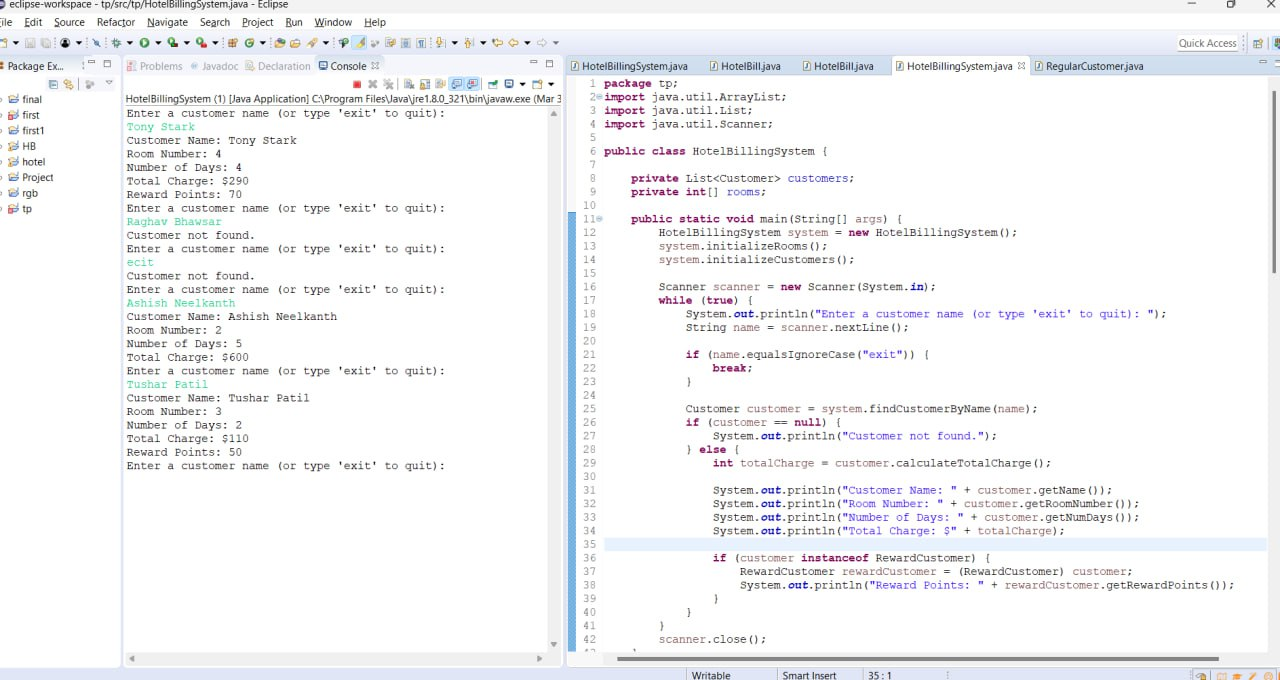
**Year**: **SEM :** 2nd year / 4th Sem

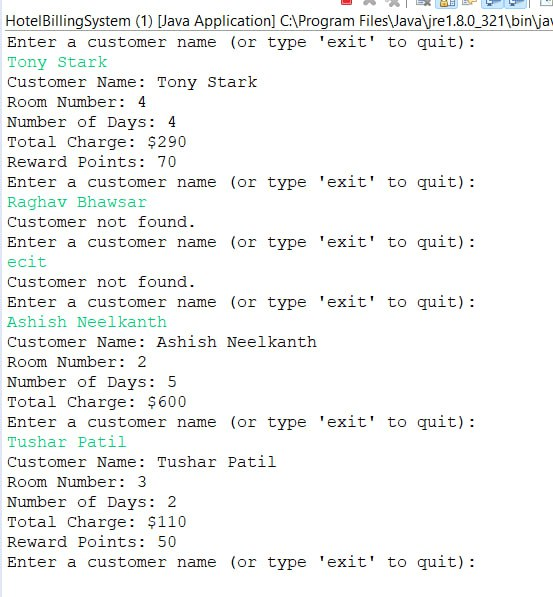
**Topic/Title**: **Hotel Billing System**

* **Introduction** : The HotelBillingSystem is a Java program that simulates a billing system for a hotel. The program uses inheritance and abstract classes to create two types of customers: RegularCustomer and RewardCustomer. RegularCustomer pays the room charge for the number of days they stayed, whereas RewardCustomer has a discount applied to their total charge based on the reward points they have accumulated. The program initializes an array of rooms and a list of customers, prompts the user to enter a customer name, and searches the list of customers for a matching name. If a matching customer is found, the program displays the customer's name, room number, number of days stayed, and total charge. If the customer is a RewardCustomer, the program also displays the number of reward points the customer has.

This program demonstrates the use of inheritance and abstract classes in Java to create a flexible billing system for a hotel. The program can be easily modified to accommodate different pricing schemes and customer types, making it useful for various types of hotels.

* **Home Screen Snap**:



* **Any other main function or Process Screen Shots**:
* 
* **Conclusion/Learning**: the HotelBillingSystem Java program provides a practical solution for managing the billing process of a hotel. The program effectively utilizes inheritance and abstract classes to create two types of customers, RegularCustomer and RewardCustomer, with different pricing schemes based on their room charge and reward points. The program demonstrates good programming practices, such as encapsulation and abstraction, and can be easily extended to accommodate additional customer types or pricing schemes. Overall, the HotelBillingSystem serves as a useful example of how object-oriented programming can be leveraged to create flexible and scalable software solutions for the hospitality industry.

**Complete Code:**

|  |
| --- |
| package tp;  import java.util.ArrayList;  import java.util.List;  import java.util.Scanner;  public class HotelBillingSystem {  private List<Customer> customers;  private int[] rooms;  public static void main(String[] args) {  HotelBillingSystem system = new HotelBillingSystem();  system.initializeRooms();  system.initializeCustomers();  Scanner scanner = new Scanner(System.in);  while (true) {  System.out.println("Enter a customer name (or type 'exit' to quit): ");  String name = scanner.nextLine();  if (name.equalsIgnoreCase("exit")) {  break;  }  Customer customer = system.findCustomerByName(name);  if (customer == null) {  System.out.println("Customer not found.");  } else {  int totalCharge = customer.calculateTotalCharge();  System.out.println("Customer Name: " + customer.getName());  System.out.println("Room Number: " + customer.getRoomNumber());  System.out.println("Number of Days: " + customer.getNumDays());  System.out.println("Total Charge: $" + totalCharge);  if (customer instanceof RewardCustomer) {  RewardCustomer rewardCustomer = (RewardCustomer) customer;  System.out.println("Reward Points: " + rewardCustomer.getRewardPoints());  }  }  }  scanner.close();  }  public void initializeRooms() {  this.rooms = new int[10];  for (int i = 0; i < rooms.length; i++) {  rooms[i] = i + 1;  }  }  public void initializeCustomers() {  this.customers = new ArrayList<>();  customers.add(new RegularCustomer("Raghav Bhawsar", 1, 3, 100));  customers.add(new RegularCustomer("Ashish Neelkanth", 2, 5, 120));  customers.add(new RewardCustomer("Tushar Patil", 3, 2, 80, 50));  customers.add(new RewardCustomer("Tony Stark", 4, 4, 90, 70));  }  public Customer findCustomerByName(String name) {  for (Customer customer : customers) {  if (customer.getName().equals(name)) {  return customer;  }  }  return null;  }  }  abstract class Customer {  protected String name;  protected int roomNumber;  protected int numDays;  protected int roomCharge;  public Customer(String name, int roomNumber, int numDays, int roomCharge) {  this.name = name;  this.roomNumber = roomNumber;  this.numDays = numDays;  this.roomCharge = roomCharge;  }  public String getName() {  return name;  }  public int getRoomNumber() {  return roomNumber;  }  public int getNumDays() {  return numDays;  }  public abstract int calculateTotalCharge();  }  class RegularCustomer extends Customer {  public RegularCustomer(String name, int roomNumber, int numDays, int roomCharge) {  super(name, roomNumber, numDays, roomCharge);  }  @Override  public int calculateTotalCharge() {  return roomCharge \* numDays;  }  }  class RewardCustomer extends Customer {  private int rewardPoints;  public RewardCustomer(String name, int roomNumber, int numDays, int roomCharge, int rewardPoints) {  super(name, roomNumber, numDays, roomCharge);  this.rewardPoints = rewardPoints;  }  public int getRewardPoints() {  return rewardPoints;  }  @Override  public int calculateTotalCharge() {  return (roomCharge \* numDays) - rewardPoints;  }  } |

**Input/Output:**

