Lab Assignment 5

Note: assignment focus on Constructor, Destructor.

Q1. Imagine a tollbooth at a bridge. Cars passing by the booth are expected to pay a 50 cent

toll. Mostly they do, but sometimes a car goes by without paying. The tollbooth keeps track of the number of cars that have gone by, and of the total amount of money collected. Model this tollbooth with a class called **tollBooth**. The two data items are a type unsigned int to hold the total number of cars, and a type double to hold the total amount of money collected. A constructor initializes both of these to 0. A member function called **payingCar()** increments the car total and adds 0.50 to the cash total. Another function, called **nopayCar()**, increments the car total but adds nothing to the cash total. Finally, a member function called **display()** displays the two totals. Make appropriate member functions const.

Include a program to test this class. This program should allow the user to push one key to count a paying car, and another to count a nonpaying car. Pushing the Esc key should cause the program to print out the total cars and total cash and then exit.

Q2. In an airline reservation system, when a family or group books tickets together, many details (such as flight number and route) remain the same. Instead of re-entering all the details for each passenger, the system can duplicate an existing ticket object.

You are required to design a **class Ticket** in C++ that demonstrates how a **copy constructor** can be used to duplicate ticket details for multiple passengers in a group booking.

Class Design

- **Data Members**
  - string passengerName
  - string flightNumber
  - string seatNumber
- **Constructors**
  - **Parameterized Constructor** – initializes passenger details.
  - **Copy Constructor** – duplicates an existing Ticket object.
- **Member Function**
  - void displayTicket() – prints ticket details.
- **Destructor**
  - Displays a message when a ticket object is destroyed (to track lifecycle).

Detailed Tasks

1. Create a **primary ticket** for the first passenger using the parameterized constructor.
2. Use the **copy constructor** to generate duplicate tickets for family/group members.
3. Modify only passenger name and seat number for new passengers (since flight number remains the same).
4. Display all ticket details for the group.
5. Show how the destructor is called for each ticket object at the end of the program.

Q3. In a company, it is essential to track how many employees are currently active in the system. Each time a new employee joins, the system should increase the employee count, and when an employee leaves (or the object is destroyed), the count should decrease automatically. You are required to design a **class Employee** in C++ that demonstrates how **static data members** can be used to track the number of live objects in a program.

Class Design

- **Data Members**
  - string name – Employee name
  - int id – Employee ID
  - static int employeeCount – Static counter for active employees
- **Constructors**
  - Parameterized constructor – Initializes employee details and increments employeeCount.
- **Destructor**
  - Decrements employeeCount and displays a message indicating which employee left.
- **Member Function**
  - void displayEmployee() – Prints the employee's name and ID.
  - static void displayCount() – Prints the current number of active employees.

Detailed Tasks

1. Create several employee objects using the parameterized constructor.
2. Display the total number of active employees after each creation.
3. Destroy some employee objects (either manually or when they go out of scope).
4. Display the total number of active employees after destruction.
5. Demonstrate that the **static counter is shared** among all objects.