

# Task-1

50 marks

Your task is to design and implement an attendance management system for a university. The system should be able to maintain attendance records for each student enrolled in a course and generate reports on student attendance.

To implement this system, you should create a `Student` class with the following members:

- `name` (a string)
- `rollNumber` (an integer)
- `attendance` (an integer)

The `name` and `rollNumber` members should be initialized in the constructor and should be declared as `const`. The `attendance` member should be initialized to 0 and should be declared as a static member of the class.

You should validate the string input for the `name` member to ensure that it contains only alphabetic characters and spaces. You should also validate the integer input for the `rollNumber` member to ensure that it is a positive integer.

The `Student` class should also have the following member functions:

- `markAttendance()`: This member function should increment the `attendance` member variable by 1. It uses the `this` pointer to access the `attendance` member variable.
- `getAttendance()`: This member function should return the value of the `attendance` member variable. It uses the `this` pointer to access the `attendance` member variable.
- `printReport()`: This member function should print a report on the student's attendance, including their name, roll number, and attendance percentage. It uses the `this` pointer to access the `name`, `rollNumber`, and `attendance` member variables.
- `static getAverageAttendance(Student students[], int numStudents)`: This is a static member function that calculates the average attendance of all the students in an array of `Student` objects. The `students` parameter is an array of `Student` objects, and `numStudents` is the number of objects in the array. This function iterates over the array, calling the `getAttendance` function on each object, and calculates the average attendance.

The `static getAverageAttendance` function is a static member function because it operates on the class as a whole, rather than on an individual object instance. This function can be called without creating an object of the `Student` class.

You should validate the string inputs for the `name` member using regular expressions. You should ensure that the input contains only alphabetic characters and spaces. If the input does not match this pattern, you should throw an exception and prompt the user to enter a valid name.

You should validate the integer input for the `rollNumber` member by checking that it is a positive integer. If the input is not a positive integer, you should throw an exception and prompt the user to enter a valid roll number.

By implementing string validation, you can ensure that the inputs for the `name` member contain only valid characters and that the input for the `rollNumber` member is in proper format. This can help to prevent errors and ensure that the system works correctly.

# Task-2

50 marks

Design and implement a program for a bus company that manages seat reservations for their buses. The program should allow customers to reserve seats on a bus, view the current seat availability, and cancel reservations if necessary. The program should have the following functionalities:

1. A Bus class that represents a bus with a certain number of seats. The Bus class should have the following member variables:

- busNumber: an integer representing the unique number of the bus
- seats: a boolean array representing the availability of each seat on the bus (true for available, false for reserved)

The Bus class should have the following member functions:

- a constructor that initializes the busNumber and sets all seats to available
- a destructor that deallocates any dynamically allocated memory
- a member function to reserve a seat, which takes an integer seat number and returns a boolean indicating whether the seat was successfully reserved or not
- a member function to cancel a reservation, which takes an integer seat number and returns a boolean indicating whether the reservation was successfully cancelled or not
- a static member function to get the total number of buses that have been created
- a constant member function to get the bus number

2. A ReservationSystem class that manages the reservations for all buses in the system. The ReservationSystem class should have the following member variables:

- buses: a vector of Bus objects representing all the buses in the system

The ReservationSystem class should have the following member functions:

- a constructor that initializes the buses vector with a certain number of Bus objects
- a destructor that deallocates any dynamically allocated memory
- a member function to reserve a seat on a bus, which takes a bus number and a seat number, and returns a boolean indicating whether the seat was successfully reserved or not
- a member function to cancel a reservation on a bus, which takes a bus number and a seat number, and returns a boolean indicating whether the reservation was successfully cancelled or not
- a constant member function to get the number of buses in the system

- a constant member function to get the total number of seats in the system
- a constant member function to get the number of available seats in the system
- a constant member function to get the number of reserved seats in the system

3. A main function that allows the user to interact with the program. The main function should have the following functionalities:

- display a menu of options for the user to choose from (e.g. reserve a seat, cancel a reservation, view seat availability, etc.)
- validate user input (e.g. ensure that the user enters a valid bus number and seat number)
- display appropriate error messages if the user enters invalid input or if a reservation or cancellation fails
- loop until the user chooses to exit the program