**Project – NeON (University Management System)**

Design an application to manage the system of a university. The structure of the university is described below:

* **Departments:** A university has many departments and each department offers several courses.
* Each department has an **HOD (Head of Department)** who manages the academic activities of the department. HOD is one of the teachers of the department.
* Each department is assigned an **IT Manager** who manages the technical facilities of the department.
* **Courses:** Each department offers several courses. One or more sections can be offered for a specific course. A maximum of 50 students can be enrolled in a course. Different departments can offer the same course.
* **Teachers:** There are several teachers of the university. Each teacher teaches a specific number of courses. A teacher can teach courses in multiple departments
* **Students:** Students get admission in one of the departments of the university. At the start of every semester, students register their courses according to the specified workload (a student can register a maximum of 5 courses in a semester).
* **Teaching Assistants:** Each course has been assigned a teaching assistant. Teaching assistants are the current students of the university.

Load the initial data of the system from a file *data.txt*. The format of the file is as follows:

* First line contains the name of the university
* Next line contains an integer D which indicates the number of departments in the university. Next lines contain information about the departments.
* First line contains department name
* Next line contains an integer C indicating the number of courses offered in this department
* C lines then follow, each containing information about a course. Each line contains course code and course name
* Next line contains an integer F indicating the number of faculty members of the department
* F lines then follow, each containing employee number and name of a faculty member. The first faculty member in this data is also the HOD of the department.
* Next line contains an integer M indicating the number of IT managers of the department
* M lines then follow, each containing employee number and name of an IT manager
* Next line contains an integer S which indicates the number of students studying in this department
* S lines then follow, each containing roll number and name of a student

|  |
| --- |
| Heavy University Lahore  2  Computer Science  3  CS100, Programming Fundamentals  CS200, Data Structures  CS300, Artificial Intelligence  2  666, Zafar Iqbal  777, Fakhar Imam  1  256, Irfan Majeed  6  1801, Muhammad Akram  1802, Umar Nadeem  1803, Zainab Irfan  1804, Hassan Afzal  1805, Ayesha Farooq  1806, Zubair Anwar  Electrical Engineering  2  EE400, Electromagnetic Theory  EE500, Multivariable Calculus  2  888, Kashif Naveed  999, Daud Khan  1  512, Sohail Ahmad  4  1861, Mohsin Iqbal  1862, Amna Khan  1863, Ali Raza  1864, Rana Saud |

**Functional Requirements**

**1. HOD (Head of Department)**

* Assign courses to teachers
* Register a course for a student
* Unregister a course for a student
* View grades of a specific course

**2. IT Manager**

* Create account of faculty members
* Delete account of faculty members
* Create account of students
* Delete accounts of students
* Maintain accounts (e.g. change username, password, etc.)

**3. Teacher**

* Manage attendance of students
* View attendance of all students
* Manage evaluations of students (e.g. enter marks, manage weightage)
* View evaluation report of a specific course displaying all evaluations of all the students of that course
* Assign grades to students

**4. Student**

* Register courses according to the specified workload
* Unregister a course
* Withdraw a course
* View marksheet of a specific course
* View attendance of a specific course
* Change password

**TODOs**

Following concepts must be reflected in your project:

* Dynamic memory allocation and deallocation
* Grow and shrink of 1D and 2D arrays
* Proper use of constructors (default, parameterized and copy) and destructors
* Memory leak and dangling pointers
* Static data members and static functions
* Inline functions
* Association, Aggregation and Composition
* Inheritance
* Polymorphism
* Abstract classes and pure virtual functions
* Multiple inheritance and diamond problem
* Initializer list
* Template functions
* Template classes
* Exception handling