Statement-

Develop a program in C++ to create a database of student's information system containing the following information: Name, roll number, Class, Division, Date of Birth, Blood group, Contact address, Telephone number, Driving license no. and other. Construct the database with suitable member functions. Make use of constructor, default constructor, copy constructor, destructor, static member functions, friend class, this pointer, inline code and dynamic memory allocation operators-new and delete as well as exception handling.

Program-

```
#include <iostream>
#include <string>
#include <stdexcept>
using namespace std;

class Student {
private:
    string name;
    int rollNumber;
    string className;
    char division;
    string dob;
    string bloodGroup;
    string contactAddress;
    string telephoneNumber;
    string drivingLicenseNo;

public:
    // Default constructor
    Student() : rollNumber(0), division('A') {}

    // Constructor with parameters
    Student(string n, int roll, string c, char d, string dob, string bg, string addr, string tel, string license)
        : name(n), rollNumber(roll), className(c), division(d), dob(dob), bloodGroup(bg),
          contactAddress(addr), telephoneNumber(tel), drivingLicenseNo(license) {}

    // Display student information
    void displayInfo() const {
        cout << "Name: " << name << "\nRoll No: " << rollNumber << "\nClass: " << className
             << " Division: " << division << "\nDOB: " << dob << "\nBlood Group: " << bloodGroup
             << "\nAddress: " << contactAddress << "\nPhone: " << telephoneNumber
             << "\nLicense: " << drivingLicenseNo << endl;
    }
};

class Admin {
private:
    Student* students[100];
    int currentIndex;

public:
    Admin() : currentIndex(0) {}

    // Add a new student
    void addStudent() {
        if (currentIndex >= 100) throw overflow_error("Database is full!");

        string name, className, dob, bloodGroup, contactAddress, telephoneNumber,
drivingLicenseNo;
        int rollNumber;
        char division;

        cout << "Enter student details: \n";
        cout << "Name: "; cin.ignore(); getline(cin, name);
        cout << "Roll Number: "; cin >> rollNumber;
        cout << "Class: "; cin >> className;
        cout << "Division: "; cin >> division;
        cout << "Date of Birth: "; cin >> dob;
        cout << "Blood Group: "; cin >> bloodGroup;
        cout << "Contact Address: "; cin.ignore(); getline(cin, contactAddress);
        cout << "Telephone Number: "; cin >> telephoneNumber;
        cout << "Driving License No.: "; cin >> drivingLicenseNo;

        students[currentIndex] = new Student(name, rollNumber, className, division, dob, bloodGroup,
contactAddress, telephoneNumber, drivingLicenseNo);
        currentIndex++;
    }

    // Display all students
    void displayAllStudents() {
        if (currentIndex == 0) {
            cout << "No students in the database.\n";
            return;
        }
        for (int i = 0; i < currentIndex; ++i) {
            students[i]->displayInfo();
            cout << endl;
        }
    }

    // Destructor to free memory
    ~Admin() {
        for (int i = 0; i < currentIndex; ++i) delete students[i];
    }
};

int main() {
    Admin admin;
    int choice;
    while (true) {
        cout << "\n1. Add Student\n2. Display All Students\n3. Exit\nEnter choice: ";
        cin >> choice;

        try {
            switch (choice) {
                case 1: admin.addStudent(); break;
                case 2: admin.displayAllStudents(); break;
                case 3: return 0;
                default: cout << "Invalid choice. Try again.\n";
            }
        } catch (const exception& e) {
            cout << "Error: " << e.what() << endl;
        }
    }
}
```