

Question 1:

```
#include <iostream>

#include <fstream>

#include <vector>

#include <string>

#include <exception>

#include <stdexcept>

using namespace std;

class User {

protected:

    string name;

    string userId;

    bool paid;

public:

    User(string n, string id) : name(n), userId(id), paid(false) {}

    virtual double calculateFare(bool ac) const = 0;

    void makePayment() {

        paid = true;

    }

    bool paymentUpdate() const {

        return paid;

    }

    string get_id() const {

        return userId;

    }

    virtual string role() const = 0;

};

class Student : public User {

public:

    Student(string n, string id) : User(n, id) {}
```

```

double calculateFare(bool ac) const {
    return ac ? 7000 : 5000;
}

string role() const {
    return "Student";
}
};

class Faculty : public User {
public:
    Faculty(string n, string id) : User(n, id) {}

    double calculateFare(bool ac) const {
        return ac ? 7000 : 5000;
    }

    string role() const {
        return "Faculty";
    }
};

class Route {
protected:
    string start;
    string end;
    int dist;
public:
    Route(string s, string e, int d) : start(s), end(e), dist(d) {}

    bool longRoute() const {
        return dist > 30;
    }

    void get_info() {
        cout << "\n" << start << " --> " << end << " | Total distance : " << dist << endl;
    }
};

```

```

class Driver {
protected:
    string name;
    string license;
public:
    Driver(string n, string l) : name(n), license(l) {}
    void get_info() {
        cout << "\nName: " << name << " | License: " << license << endl;
    }
};

class Seat {
protected:
    bool booked;
    string role;
public:
    Seat(string r) : booked(false), role(r) {}
    bool available() {
        return !booked;
    }
    string get_role() {
        return role;
    }
    void booking() {
        booked = true;
    }
};

class Vehicle {
protected:
    string id;
    Driver* d;
    Route* r;

```

```

    bool AC;

    vector<Seat> seats;

public:
    Vehicle(string id, Driver* d, Route* r, string t, bool a) : id(id), d(d), r(r), AC(a) {
        int seatcount = (t == "Bus") ? 52 : 32;
        for (int i = 0; i < seatcount; ++i) {
            seats.push_back((i % 2 == 0) ? Seat("Student") : Seat("Faculty"));
        }
    }

    bool bookSeat(User* u) {
        if (!u->paymentUpdate()) {
            throw runtime_error("Payment not completed!");
        }

        for (int i = 0; i < seats.size(); i++) {
            if (seats[i].available() && seats[i].get_role() == u->role()) {
                seats[i].booking();
                return true;
            }
        }

        throw runtime_error("No available seat for your role!");
    }

    void get_info() {
        cout << "\nVehicle id: " << id << "\tDriver: ";
        d->get_info();
        cout << "\tRoute: ";
        r->get_info();
    }

    bool isAirConditioned() {
        return AC;
    }
};

```

```

class Bookingg {
protected:
    User* u;
    Vehicle* v;
public:
    Bookingg(User* u, Vehicle* v) : u(u), v(v) {
        v->bookSeat(u);
    }
    void show_booking() {
        cout << "Booking Successful | User: " << u->get_id();
        cout << " | Role: " << u->role();
        cout << " | Vehicle: ";
        v->get_info();
    }
};

int main() {
    try {
        Route r1("Nazimabad", "FAST", 30);
        Route r2("Defence", "FAST", 35);
        Driver d1("Rayyan", "L-1234");
        Driver d2("Rayy", "L-5678");
        Vehicle v1("V001", &d1, &r1, "Bus", true);
        Vehicle v2("V002", &d2, &r2, "Coaster", false);
        Student s1("Rayyan", "S001");
        Faculty f1("Nida Munawarr", "F001");
        cout << "Fare for Rayyan (AC): Rs. " << s1.calculateFare(true) << endl;
        s1.makePayment();
        Bookingg b1(&s1, &v1);
        b1.show_booking();
        f1.makePayment();
        Bookingg b2(&f1, &v2);
    }
}

```

```
b2.show_booking();

ofstream file("bookings.txt");

file << "User: " << s1.get_id() << ", Vehicle: V001\n";

file << "User: " << f1.get_id() << ", Vehicle: V002\n";

file.close();

}

catch (runtime_error& e){

    cerr << "Booking error: " << e.what() << endl;

}

catch (exception& e){

    cerr << "General error: " << e.what() << endl;

}

return 0;

}
```

```
Fare for Rayyan (AC): Rs. 7000
Booking Successful | User: S001 | Role: Student | Vehicle:
Vehicle id: V001      Driver:
Name: Rayyan | License: L-1234
Route:
Nazimabad --> FAST | Total distance : 30
Booking Successful | User: F001 | Role: Faculty | Vehicle:
Vehicle id: V002      Driver:
Name: Rayy | License: L-5678
Route:
Defence --> FAST | Total distance : 35

-----
Process exited after 0.2666 seconds with return value 0
Press any key to continue . . .
```