National University
of computer and emerging sciences

CL-1004

Object Oriented Programming- Lab

Spring' 2023

BS-SE

Lab Sessional

1 Hour

Note: Plagiarism (either from internet or from someone else) will result in zero marks.

Ravka Transportation System:

We are building a transport system that includes vehicles, routes, and passengers. The system should be able to handle multiple vehicles and routes, each with their own schedules, and multiple passengers that can book rides on these vehicles and routes.

A manager opens the system and adds the East Ravka to West Ravka route to the list of routes in the transportation system, specifying the desired departure and arrival times. He can also specify what type of vehicle is available for each route. The system creates a Route instance for the East Ravka to West Ravka train route. It also creates a Time instance for different departure and arrival times and adds it to the Schedule instance for the route. (Add at least 3 different routes)

The manager can also sort the routes by comparing them on the basis of various criteria such as type of vehicle or desired departure and arrival times.

Later, suppose a passenger named Mal wants to travel from East Ravka to West Ravka on a bike. When Mal wants to book a ride, booking instance is created for which the system searches for available transports on the East Ravka to West Ravka route that meet his requirements such as type of vehicle and desired departure and arrival times. If requirements are not fulfilled, booking instance is not created.

Mal can see schedule of routes for any time, he can enter departure time into the system and the system would display all the bookings of all the clients at that time. He can book multiple rides as well at different departure time (Make him book at least 2 rides for different times.). He can cancel the booking as well, booking instance for that ride would be removed from Mal's profile.

On the day of the ride, John boards the train and travels from East Ravka to West Ravka. After the ride is completed, booking is Mal is removed.

Note:

- Identify all the classes, their relations (inheritance, aggregation and composition) and all the necessary attributes and functions.
- Identify cases of recursion and operator overloading (there are hidden somewhere in the case study).
- All the necessary functions (default constructor, parameterized constructor, copy constructor, getter and setters) should be implemented.
- Create a driver function (main function), and call the functions in a way that realizes the above scenario. Do this for 2 customers, having different values.
- Failing to identify all the covered concepts will lead to deduction of marks.

Submission Details:

1. Save all the .h and .cpp files.

2. Make zip file of all the header and source files (Do not create .rar file) with roll no and lab no. e.g. i22-XXXX_OOP_SessionalExam.zip.

3. Submit the zip file on google class room.