Exam 3 – Programming Lab

CSC/DSCI 1301 – Principles of CS/DS I

Monday, April 14th, 2024

Lab Exam Policy

This is a closed book exam! You may only have your cheat sheet and your IDE open. You may not use an internet browser to research the solution. There is only 1 programming question. Please read the question carefully. You will have 1 hour and 40 minutes to complete this exam. Note: Please leave time to submit your code on iCollege. The iCollege submission window will close sharply at the end of the lab session.

Section	Grade	Points
Programming		40
Total		**40**

Deliverables

For this exam you will need to submit the python file containing your code to the iCollege assignment by the end of the lab. Please name your files as follows:

- Python Files
    - lastname_firstname_filename.py
    - For example: **hawamdeh_faris_train_planner.py**

Program: train_planner.py

For this exam, you will need to write a program that can keep track of the composition of a train and its cargo. Your program will compute the length and speed of potential train configurations. You will need to write three **classes** to represent the **train**, the **cars** that carry the cargo and the **locomotive** that pulls the train.

Class: Locomotive

A **locomotive** or **engine** is the vehicle that provides the pulling or pushing power for a train. The **locomotive** class should store information about the specifications of each **engine**: length, maximum payload, and maximum speed. The average length of a locomotive is 23 meters.

- **Length** – The length of the locomotive/engine in **meters.**
- **Maximum Payload** – The maximum weight the locomotive/engine can pull in **tons**.
- **Maximum Speed** – The maximum speed the train can travel with no cargo in **miles per hour**.

Your locomotive class must implement **methods** to get the **length**, **maximum payload,** and **maximum speed** for all objects of the locomotive class:

- **get_length**() – Returns the length of the locomotive/engine in **meters**.
- **get_maximum_payload**() – Returns the maximum payload of the locomotive/engine in **tons**.
- **get_maximum_speed**() – Returns the maximum speed of the locomotive in **miles per hour**.

Class: Railcar

A **railroad** car or **railcar** is an unpowered vehicle used for carrying cargo or passengers. The railcar class should store information about the specifications of each railcar: length, type, minimum weight, maximum weight, and capacity. The average length of a rail car is 20 meters.

- **Length** – The length of the railcar in meters.
- **Cargo Type** – The type of cargo being carried:
  - Passenger
  - Freight
- **Minimum Weight** – The weight of the railcar when at 0% capacity in **tons**.
- **Maximum Weight** – The maximum weight when at 100% capacity in **tons**.
- **Capacity** – The amount of cargo in the railcar as a fraction, (0.0 – 1.0), of total capacity (Max Weight – Min Weight).

Your Railcar class must implement **methods** to get the **length**, **cargo type**, and **current weight** of a railcar object:

- **get_length()** – Returns the length of the railcar in meters.
- **get_cargo_type()** – Returns the type of cargo.
- **get_weight()** – returns the current weight of the railcar in tons.

Class: Train

A **train** consists of **at least 1 locomotive** and **0 or more railcars**. The **train** class should consist of two lists: locomotives and cars. The **speed** of the train is dependent on the maximum speed and the current payload of the railcars. The train's **speed decreases** linearly as its payload increases up to its maximum payload. The **minimum speed** of the train is 50% of its maximum speed when at its maximum payload.

The following attributes of the train class should be calculated from the list of locomotives and cars:

- **Maximum Payload** – The **maximum payload** of the train is the sum of the maximum payload of its locomotives.
- **Maximum Speed** – The **maximum speed** of the train is the maximum speed of the slowest locomotive pulling 0 weight.
- **Length** – The sum of the lengths of all locomotives and railcars.
- **Payload** – The sum of the weight of all railcars.
- **Speed** – The speed of the train given its current weight.

Your Train class must implement methods to get the **length**, **payload**, and **speed** of the train in its current configuration:

- **get_length()** – Returns the length of the train in **meters**.
- **get_payload()** – Returns the weight of the cargo in **tons**.
- **get_speed()** – Returns the speed of the train in **miles per hour**.

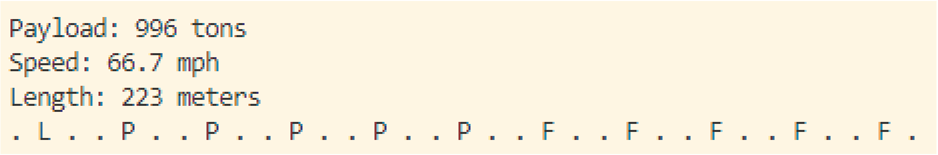
Your train class must implement methods to add and remove railcars and locomotives from the train:

- **add_railcar (railcar)** – Adds the input railcar to the end of the train **if the railcar does not cause the train to exceed its maximum payload**.
- **remove_railcar()** – Removes the railcar at the end of the train.
- **add_locomotive (locomotive)** – Adds the input locomotive to the locomotive list.
- **remove_locomotive ()** – removes the last locomotive from the locomotive list **if the locomotive is not the only locomotive in the train.**

You train class must implement a method to print out the composition of the train:

- **print_train()** – Prints the following information about the planned train:
    - **Payload** – The weight of the cargo carried by all railcars
    - **Speed** – The speed of the train given the weight of its railcars
    - **Length** – Length of the train including all railcars and locomotives
    - **Composition** – 1 Letter abbreviation for the locomotives and type of rail cars. See Example Output.

Example Output



```
Payload: 996 tons
Speed: 66.7 mph
Length: 223 meters
. L . . P . . P . . P . . P . . P . . F . . F . . F . . F . . F .
```