

Exam 3 – Programming Lab

CSC/DSCI 1301 – Principles of CS/DS I

Tuesday, April 16th, 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_ship_planner.py**

Program: ship_planner.py

For this exam, you will need to write a program that can keep track of the composition of a container ship and its cargo. Your program will compute the speed and draught of potential container ship configurations. You will need to write two **classes** to represent the **container ship**, the **containers** that carry the cargo.

Class: Container

An **intermodal container** or **shipping container** is a standardized container design that can be used across different modes of transport, from **ship** to **rail** to **truck**, without unloading and reloading their cargo. A container can come in two standard sizes: 1 TEU and 2 TEU. The **container** class should store information about the specifications of each container: **size** and **cargo type**.

- **Size** – The size of the container: 1 TEU or 2 TEU
- **Cargo Type** – The type of cargo being carried:
 - Frozen Food – FF
 - Consumer Goods – CG
 - Processed Goods – PG
 - Raw Materials – RM
 - Industrial Equipment – IE

Your container class must implement **methods** to get the **size** and **cargo type**:

- **get_size()** – Returns the size of the container in TEU.
- **get_cargo_type()** – Returns the type of cargo.

Class: Container Ship

A **container ship** is a type of cargo ship that carries all its cargo in truck-sized containers. A container ship's capacity is measured in twenty-foot equivalent units (TEU). Typically, a container ship's cargo is split between 20-foot (1-TEU) and 40-foot (2-TEU) containers. The constructor for your container ship class will take the following parameters as input:

- **Maximum Capacity** – The maximum number cargo capacity of the container ship in **TEUs**.
- **Maximum Speed** – The maximum speed the container ship can sail with no cargo in **nautical miles per hour (knots)**
- **Minimum Draft** – The minimum depth of the container ship with no cargo in **meters**.
- **Maximum Draft** – The maximum depth of the container ship with full cargo in **meters**.

Your container ship class should store a list of all the containers assigned to the cargo ship into an attribute called **containers**. The **draft** of the ship is dependent on the current cargo of the ship and its maximum capacity. The ship's **draft increases** linearly as its capacity increases up to its maximum capacity. The **speed** of the ship is dependent on its maximum speed and its current draft. The ship's **speed decreases** linearly as its draft increases up to its maximum draft. The **minimum speed** of the ship is 50% of its maximum speed when at its maximum draft.

The following attributes of the container ship class should be calculated from the list of containers:

- **Cargo** – The amount of cargo assigned to the container ship in TEUs.
- **Draft** – The depth of the container ship given its current cargo and maximum capacity.
- **Speed** – The speed of the container ship given its current draft.

Your container ship class must implement **methods** to get the **capacity**, **draft**, and **speed** for all objects of the container ship class:

- **get_cargo()** – Returns the capacity a **TEUs**.
- **get_draft()** – Returns the draft of the container ship in **meters**.
- **get_speed()** – Returns the speed of the container in **knots**.

Your container ship class must implement methods to add and remove containers from the ship:

- **add_container(container)** – Adds the input container to the top of the container ship **if the container does not cause the ship to exceed its maximum capacity**.
- **remove_container()** – Removes the container at the top of the container ship.

Your container ship class must implement a method to print out the composition of the ship:

- **print_ship()** – Prints the following information about the planned cargo of the ship:
 - **Cargo** – The amount of cargo that has been filled in **TEUs**.
 - **Draft** – Depth of the containership given its current cargo in **meters**.
 - **Speed** – The speed of the container ship given its current draft in **knots**.
 - **Composition** – 2 Letter abbreviation for the cargo for each container split across 4 rows. (Containers can be stacked to a maximum height of 4). Cargo is loaded bottom up. See Example Output.

Example Output

```
Cargo: 22 TEUs
Draft: 29.7 meters
Speed: 11.5 knots
```

```
[ CG ] [ IE ] [ PG ]
[ PG ] [ PG ] [ PG ] [ RM ]
[ RM ] [ CG ] [ IE ] [ PG ]
[ PG ] [ PG ] [ RM ] [ PG ]
```