ACS-1904 W2024
Lab #5
Due by Friday, February 9 at 11:59 pm

- Combine/compress all of the Java files for this programming question into a .zip file and submit the .zip file via Nexus. The Nexus drop box closes at 11:59 pm.
    - Code submitted in any other way will not be accepted.
    - Include your name and student number in each file as a comment.

Develop a set of classes that form a **hierarchy** for ships where a `Ship` can be a `Tanker` and where a `Ship` can be a `CruiseShip`.

Include fields that are common to `Tanker` and `CruiseShip` in a `Ship` class: Strings `name` (the name of the ship), `flag` (the country of registration), and int `year` (the registration year of the ship). The `Ship` class also has a `getName()` method that returns a catenation of the ship's name and its flag (e.g. Evergreen, Panama). Override the `toString()` method to return a string that includes the ship's name and registration year.

Include the following fields for each of the subclasses: choose appropriate data types

`Tanker`: `capacity` (maximum carrying capacity in litres), `length` (in meters)
`CruiseShip`: `passengers`(number of passengers), `cruiseLine`(the name of the cruise line operating the ship

All classes must have no-arg and full-arg constructors. Getters and setters are optional, only methods needed to produce the output are required. Have both `Tanker` and `CruiseShip` classes override the `getName()` method in the following manner:

`Tanker`:	*`name capacity(L) length(m)`*	e.g. **Ranger, 25000L, 450m**
`CruiseShip`:	*`cruiseLine name, passengers`*	e.g. **MSV, Aurora, 5453**

Note: the units i.e. L and m in the above examples are part of the output but not part of the value of the fields.

Notes:

- Use the Practitioner example from our notes/textbook to guide you.
- Ensure that all fields in the superclass have protected access.

Demonstrate the features of your hierarchy with a class named Ships that instantiates one object of each type and adds them to an ArrayList of type <`Ship`>

Use a for-each (enhanced for) loop to display each of the objects by calling their `getName()` method.

Print the second element in the ArrayList using its `toString()` method.

Sample output:

```
Evergreen, Panama
Ranger, 250000L, 450m
MSV, Aurora, 5453

Ranger: 1997

end of program
```

Submit your Lab5.zip file that includes all lab files (Ship.java,Tanker.java, CruiseShip.java, Ships.java) via Nexus.