Due Date: Sunday, December 5, 2021, 23:59
Total Points: 5

You are given a base class encapsulating the concept of a ship, called *Ship*. The class has three data members, the name of the ship (a string), the year that the ship was built (an integer), and the type of vessel (string). The class interface is given in the header file "ship.h" and has the following definitions.

```
class Ship {
  private:
      string shipName;
      int shipYear;
      string vesselType;
  public:
      Ship(string name, int year): shipName{name}, shipYear{year}{
            vesselType = "Ship";
      }
      string getShName() const{
            return shipName;
      }
      int getshYear() const{
            return shipYear;
      };
      string getvesselType(){
            return vesselType;
      }
      void setvesselType(string type){
            vesselType = type;
      }

      virtual void print() const ;      // output ship info
}; // end class Ship
```

The *Ship* base class has two derived classes (i.e., subclasses), a *Cruise* class and a *Cargo* class. The *Cruise* class has an additional field, the maximum number of passengers (an integer). Include accessor(getter) and mutator(setter) functions for the *Cruise* class's additional data member. The *Cargo* class has one additional field, the cargo capacity in tonnage (an integer). Include accessor(getter) and mutator(setter) functions for the *Cargo* class's additional data member. Both classes have a constructor that is passed parameters to initialize the class's data members. The interface definitions of each one of the classes are given in the header files "cruise.h" and "cargo.h", respectively.

a. Implement the *print* function in the *Ship* class.
b. Implement the *Cruise* functions.
c. Implement the *Cargo* class functions.

You can implement all the classes' functions in one file, say "Shipcollection.cpp" or provide the implementations of each class functions in separate files as "ship.cpp", "cruise.cpp", and "cargo.cpp. All the header files will be propagated to your Work area automatically on Vocareum, along with the driver program RA8pro.cpp.

Vocareum Automatic Grading
- Upload your implementations file(s) to Vocareum.
- The driver program "RA8prog.cpp" will be used to test your implementations. The grading of the implementation is based on the expected output, as shown below.

```
Vessel 1 Cruise:
Name: Los Angles   Year: 1990   Maximum Passengers: 2500

Vessel 2 Cargo:
Name: Georgia   Year: 2000   Capacity (in tons): 400

Vessel 3 Ship:
Name: Houst   Year: 1950

Vessel 4 Ship:
Name: Tiger   Year: 1970

Vessel 5 Cruise:
Name: Boston   Year: 1985   Maximum Passengers: 1200

Vessel 6 Cargo:
Name: Alabama   Year: 2010   Capacity (in tons): 600
```

Submission Guidelines

- Please upload your implementations to Vocareum as one file "shipcollection.cpp" or in separate files one for each class implementation.
- **Submissions after the due date are accepted with a fixed penalty of 25%. No submission is accepted after Tuesday 11:59 pm, December 7, 2021.**

Grading Table

Item	Points
Compiles Successfully	1
Output information for all objects	4
Total	5