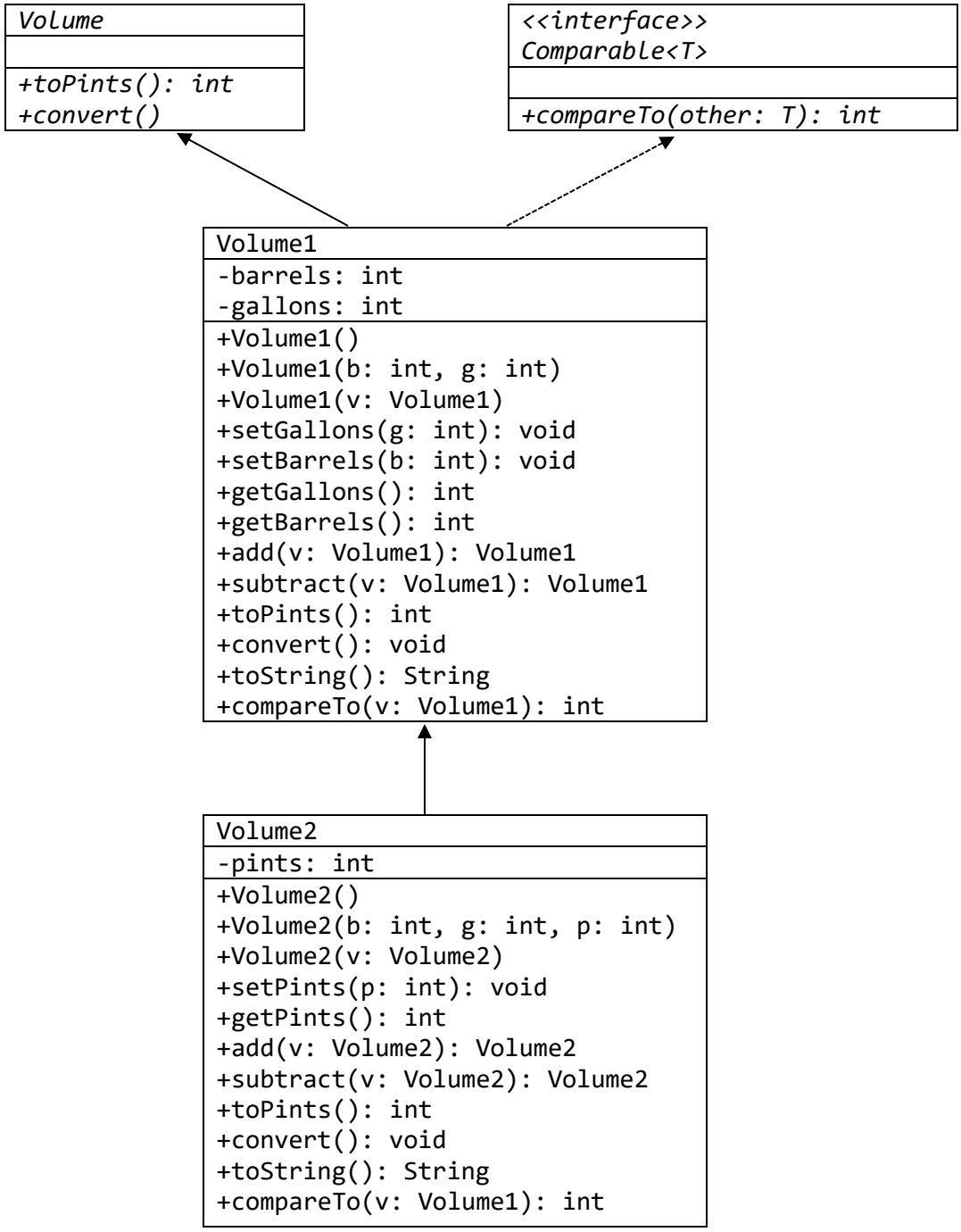
CS1760 ADV OBJ-ORIENTED PROG & DESIGN

Programming Assignment 1

Create a Java class hierarchy to represent oil volumes in barrels, gallons and pints using the UML diagrams below. Create abstract class **Volume** and classes **Volume1**, **Volume2**.



Method Notes

- **add and subtract()** are invoked as **a.add(b)** and **a.subtract(b)**. Both **add()** and **subtract()** must return **Volume1**s or **Volume2**s that have been **convert**ed
- **toPints()** – returns the **this** object as a measurement in pints
- **convert()** – converts **this** into a form where the number of gallons is greater than or equal to 0 and less than 42, number of pints is greater than or equal to 0 and less than 8
- You can add extra methods if needed or exclude an inherited method that is not overridden.

Testing
Write a test file, **Assign1** that does **exactly** the following:

- Declare **Volume1** references a, b, c, d, e, f, g, h
- Declare **Volume2** references w, x, y, z
- Instantiate objects (with **new**) as follows:
  - a = 114 barrels, 37 gallons
  - b = 56 barrels, 41 gallons
  - w = 57 barrels, 38 gallons, 6 pints
  - x = 56 barrels, 41 gallons, 7 pints
  - c, d as **Volume1** objects (no parameters)
  - y, z, e, f, g, h as **Volume2** objects (no parameters)
- Carry out the following calculations: (use **Add** and **Subtract** methods, no casting)
  - c = a + b
  - d = a – b
  - y = w + x
  - z = w – x
  - e = a + w
  - f = a – w
  - g = x + b
  - h = x - b
- Create a **Volume** array {a, b, c, d, e, f, g, h, w, x, y, z}
- Print the contents of the array
- Sort the array
- Print the contents of the array
- Print **`b.compareTo(x)`** (should be negative integer)
- Print **`x.compareTo(b)`** (should be positive integer)
- Print a in pints
- Print w in pints

Notes
1. No GUI interface is necessary – display using **System.out**.
2. When defining **compareTo()**, you should return –1 if the "this" object is less than the parameter object, return 0 if both objects are equal, and 1 if the "this" object is greater than the parameter object. Your program must be able to compare objects of the same type or different types.
3. The method **convert()** is used to ensure correct format for the numbers. It should be invoked within the **Volume1** and **Volume2** classes, for example when adding.
4. To sort the array, use the sort method in the **Arrays** class (see **People** example).
5. **When overriding methods in Volume2, you should use calls to the corresponding superclass methods where possible**.
6. 1 barrel = 42 gallons, 1 gallon = 8 pints. You can assume that we are not dealing with negative total volumes when instantiating objects or subtracting.

Turning in the Assignment
Upload all **java** and **class** files to Canvas by **Monday 31 January at 4pm**.You should zip all the files into one file before you upload – include whole directory structure if you made a package. Also turn in a screenshot of your program running and a copy of the Assignment Information Sheet.