**Problem Set 4, Part I**

***Please start your answer to each problem on a new page, as we have done below!***

**Problem 1: Understanding and using inheritance**

**1-1)** Truck, Vehicle

**1-2)** Yes, because getMileage() is defined in the Vehicle class and is not overridden by Truck, the superclass of TractorTrailer. Truck inherits all methods of Vehicle, and TractorTrailer inherits all methods of Truck, so it should be able to make the method call to getMileage().

**1-3)** *Add your definition of the Limousine class below:*

```
public class Limousine extends Automobile {
      private boolean hasSunRoof;  // true or false
      private int numChampagne;  // a non-negative integer

      public Limousine(String make, String model, int year, int numSeats,
         int hasSunRoof, int numChampagne) {
            if (numSeats <= 0 && numChampagne <= 0) {
                  throw new IllegalArgumentException();
            }
            super(make, model, year, numSeats, true);
            this.hasSunRoof = hasSunRoof;
            this.numChampagne = numChampagne;
      }

      public int getHasSunRoof() {
            return this.hasSunRoof;
      }

      public int getNumChampagne() {
            return this.NumChampagne;
      }

      public String toString() {
            String str = this.getMake() + " " + this.getModel();
            str += " (seats up to "+(this.getNumSeats()-2);
            str += " customers)";
            return str;
      }

}
```

**Problem 2: Inheritance and polymorphism**

**2-1)** The equals() method that Zoo overrides comes from the Object class. Zoo does not explicitly extend a class, but it implicitly extends the Object class, which is at the top of class hierarchy. The Object class defines the default equals() method, which is inherited by Zoo, and Zoo overrides it and has its own equals() method.

**2-2) a, b, x, y, c**

**2-3)**

| **which println statement?** | **which method is called?** | **will the call compile (yes/no?)** | **if the call compiles, which version of the method will be called?** |
|---|---|---|---|
| first one | one() | yes | the Yoo version |
| second one | two() | yes | the Woo version |
| third one | three() | no | N/A |
| fourth one | equals() | yes | the Zoo version |
| fifth one | toString() | yes | the Woo version |

**2-4)**

```
public double avg() {
      double average = (this.getA() + this.t + this.u) / 3.0;
      return average;
}
```

**2-5)**

**a) No. Woo is not a superclass of Too but is of the same level of class hierarchy as Too.**

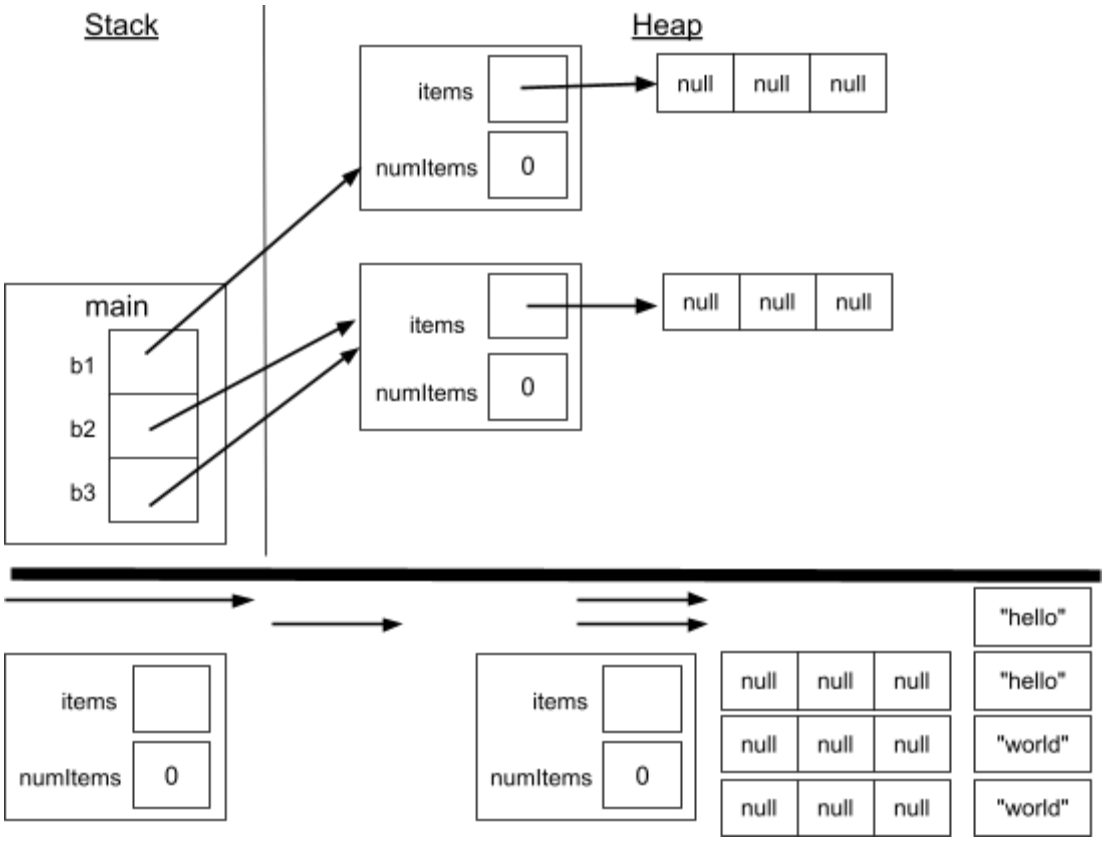
**b) Yes. Zoo is the immediate superclass of Woo, and Woo objects are of a certain type of Zoo objects.**

**c) Yes. Zoo is a superclass of Yoo, and Yoo objects are of a certain type of Zoo objects.**

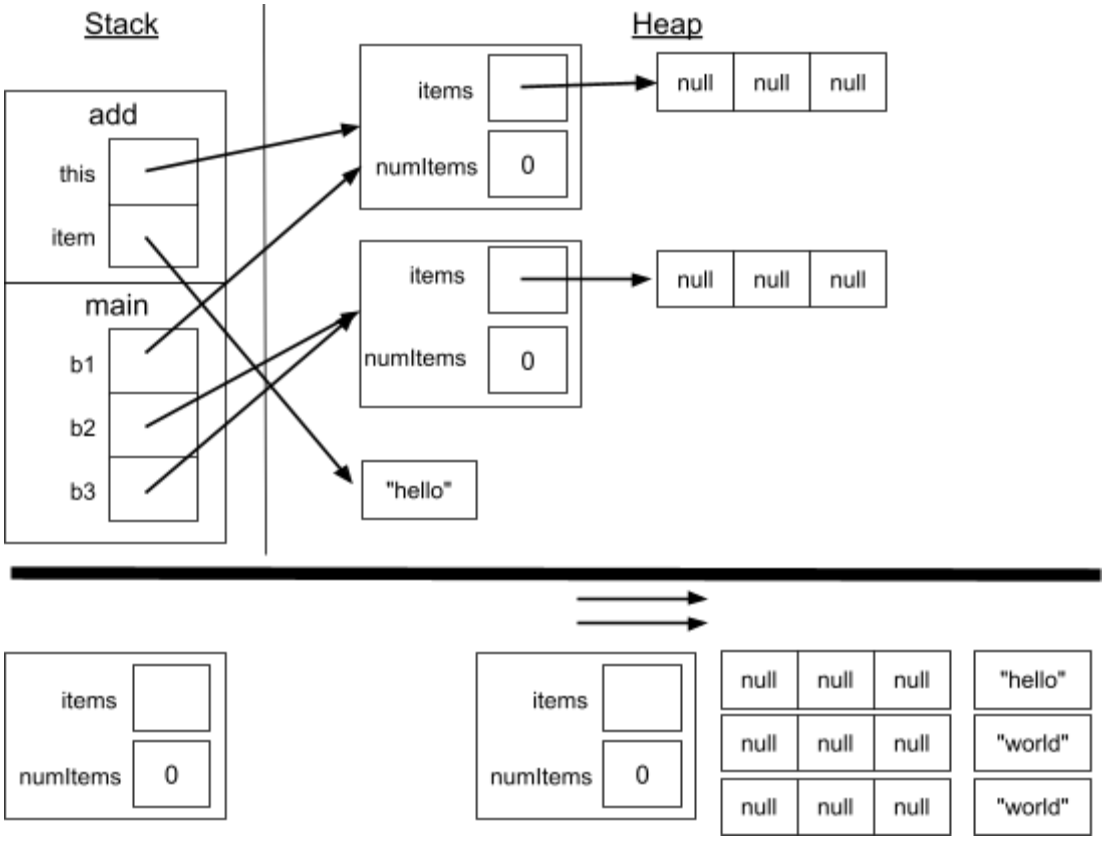
**d) No. Too is an immediate subclass of Zoo, so Too classes cannot perfectly define all possible Zoo objects.**

**Problem 3: Memory management and the `ArrayBag` class**

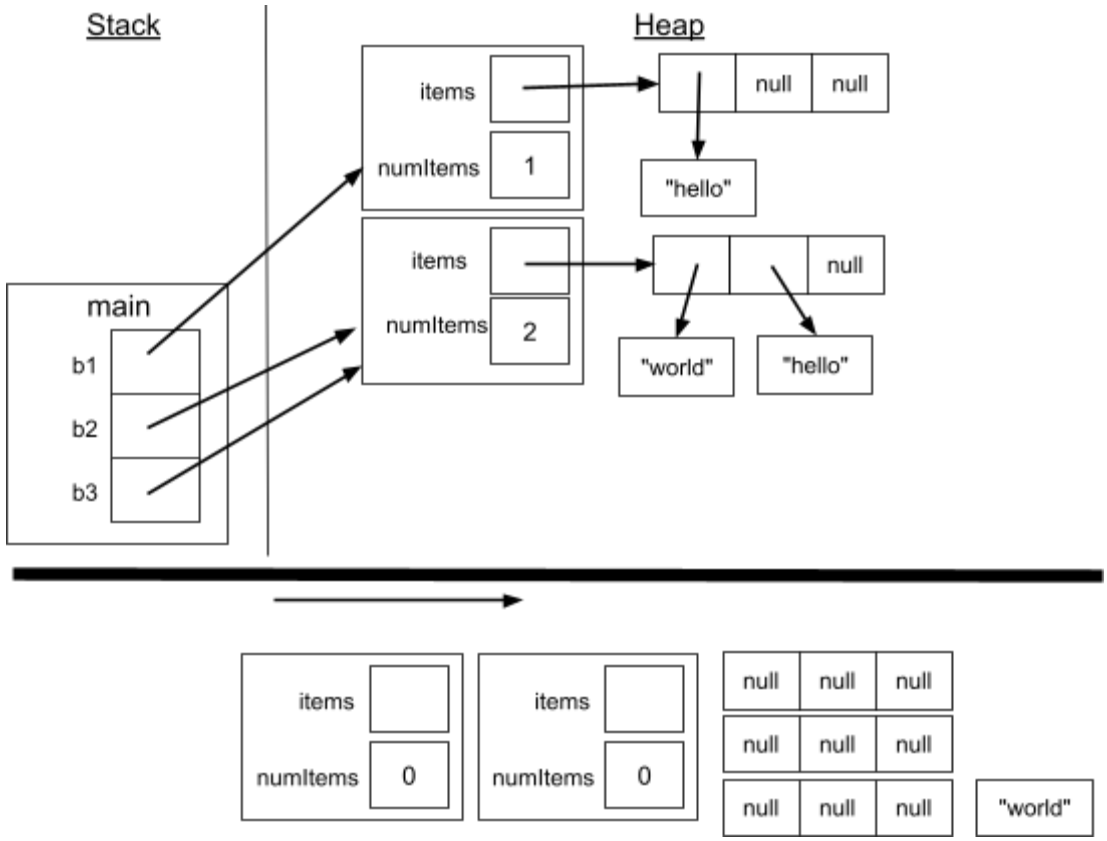
**3-1)**




**3-2)**



**3-3)**




**3-4)**

```
{hello}
{world, hello}
{world, hello}
```