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) Automobile is the superclass of the Taxi class

1-2) Yes, you can make that call getNumSeats() because it is inherited from the automobile class. Therefore, it will be available to be called.

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

```
public class Limousine extends Automobile {
      private boolean hasSunRoof;
      private int chillBottles;
      private String color;

      public Limousine(String make, String model, int year,
      boolean hasSunRoof, int chillBottles, String color) {
            super(make, model, year, 4, false);
            if (chillBottles < 0) {
                  throw new IllegalArgumentException();
            }
            if (color.equals(“”)) {
                  throw new IllegalArgumentException();
            }
            this.hasSunRoof = hasSunRoof;
            this.chillBottles = chillBottles;
            this.color = color;
      }

      // Accessors
      public boolean getSunRoof() {
            return this.hasSunRoof;
      }

      public int getChillBottles() {
            return this.chillBottles;
      }

      public String getColor() {
            return this.color;
      }

      public String toString() {
            return (this.getColor() + “ “ + this.getMake() + “ ” +
            this.getModel() + “ (seats up to “ + (this.getNumSeats() - 2) +
            “ customers)”);
      }
}
```

Problem 2: Inheritance and polymorphism

2-1) The Zoo's equal() method overrides the inherited method from the Object class. Every class implicitly extends the Object class, and because Zoo doesn't extend anything it inherits equals from Object.

2-2)
```
public Woo(int a, String b, int x, int y, int sum, String c) {
      super(a, b, x, y);
      this.sum = sum;
      this.c = 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	y.one(10)	yes	the Yoo version
second	y1.two()	yes	the Woo version
third	y1.three(12.5)	no	N/A
fourth	y1.equals(y1)	yes	the Zoo version
fifth	toString(y1)	yes	the Woo version

2-4)
```
public double avg() {
      return ((this.t + this.u + this.getA()) / 3.0);
}
```

2-5)

a) Won't compile, because the Woo class is on a different branch of inheritance and therefore a Too object would not have sufficient functionality to satisfy the needs of a Woo object.

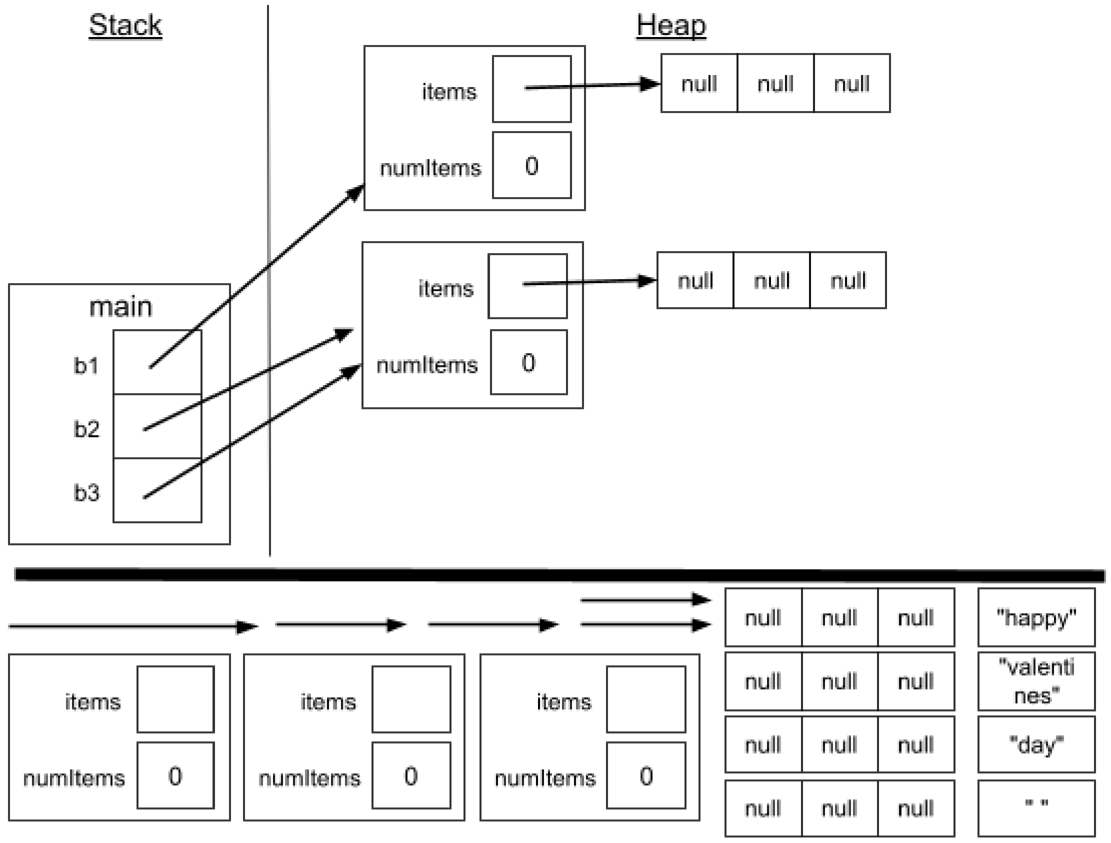
b) Will compile, because the Woo class is a subclass of Zoo it will have all of the same functionality of a Zoo class and will cause no problems.

c) Will compile, because the Yoo class is a subclass of Woo which is a subclass of Zoo it will have all of the same functionality of a Zoo class and will cause no problems.

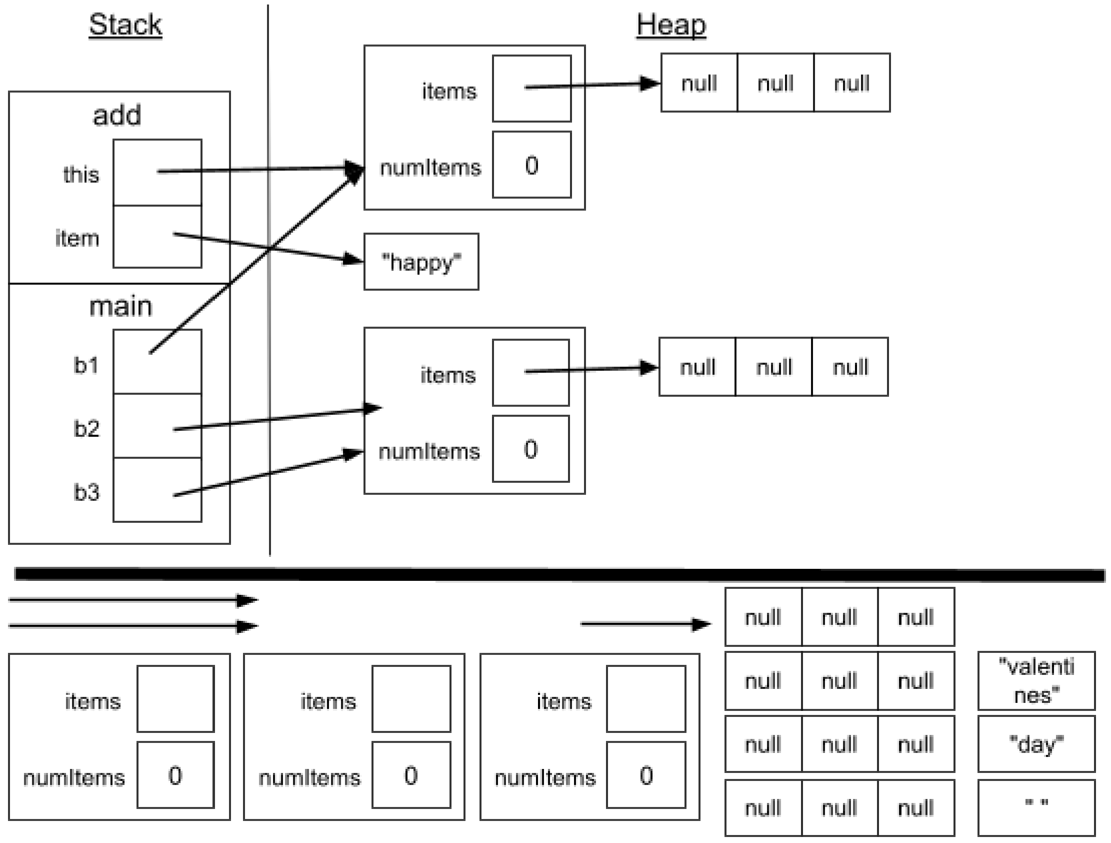
d) Won't compile, because the Too class is a subclass of Zoo, Zoo won't have sufficient functionality to be a Too class and will cause problems.

Problem 3: Memory management and the `ArrayBag` class

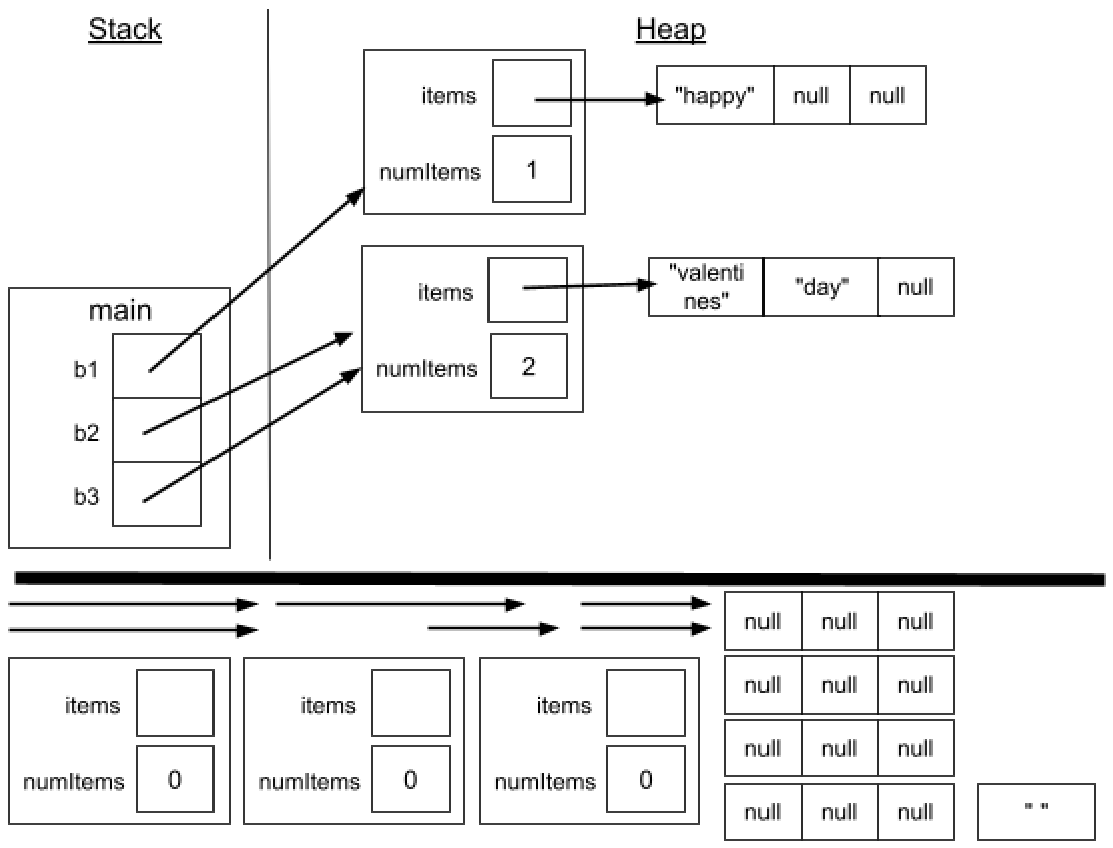
3-1)



3-2)



3-3)



3-4)

```
{happy}
{valentines, day}
{valentines, day}
```