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) The automobile class

1-2) It is possible to make the call with the taxi object. Taxi inherits the superclass methods of the automobile class, which supplies it with the method.

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

```
public class Limousine extends Automobile {
	private boolean sunRoof;
	private int champagneAmount;
	private String color;

	// constructor method
	public Limousine(String make, String model, int year, int seatTotal,
boolean sunRoof, int champagneAmount, String color) {
		super(make, model, year, seatTotal - 2);
		this.sunRoof = sunRoof;
		this.champagneAmount = champagneAmount;
		this.color = color;
	}

	// getter methods
	public boolean getSunRoof() {
		return sunRoof;
	}

	public int getChampagneAmount() {
		return champagneAmount;
	}

	public String getColor() {
		return color;
	}

	// toString method (overrides the automobile toString method)
	public String toString() {
		String result = “”;
		result += this.color + “ ” + this.make + “ ” + this.model +
“ (seats up to ” + this.seatTotal + “ customers)”;
		return result;

	}

}
```

Problem 2: Inheritance and polymorphism

2-1) the equals() method that the Zoo class overrides comes from the Object class that the Zoo class inherits from. All classes in Java has a superclass of an Object class.

2-2)

```
public Yoo() {
      super();
      String c = “”;
      int sum = 0;
}
```

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	
fourth one	equals()	yes	the Zoo version
fifth one	toString()	yes	the Yoo version

2-4)

```
public double avg() {
      int sum = t + u + getA(); // getA() is an accessor method that gets the field “a” from the superclass, which is the Zoo class
      int integerFields = 3;
      return (double) sum / integerFields;
}
```

2-5)

a) This is not allowed because in the inheritance tree, they are not on the same side. This means that Woo is not a superclass of Too.

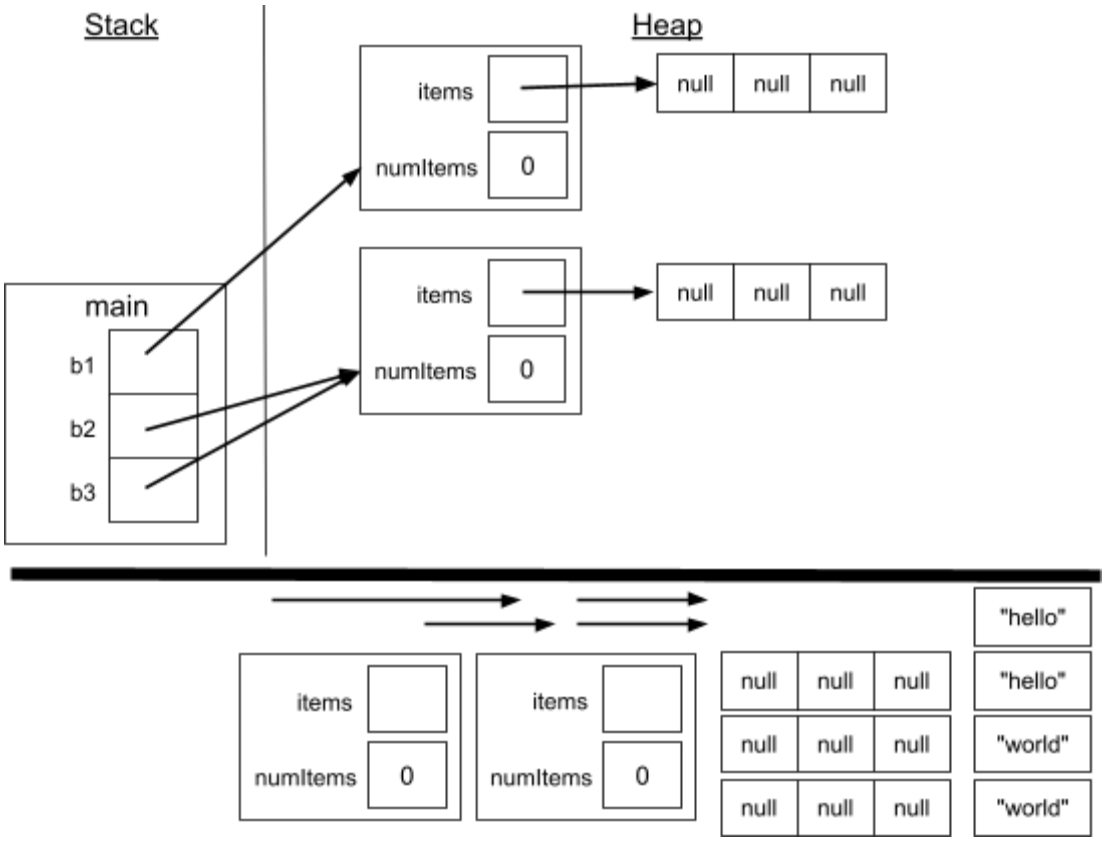
b) This is allowed because Zoo is a superclass of Woo, so that allows the object to use the suclass reference variables.

c) This is allowed because Zoo is a superclass of Yoo.

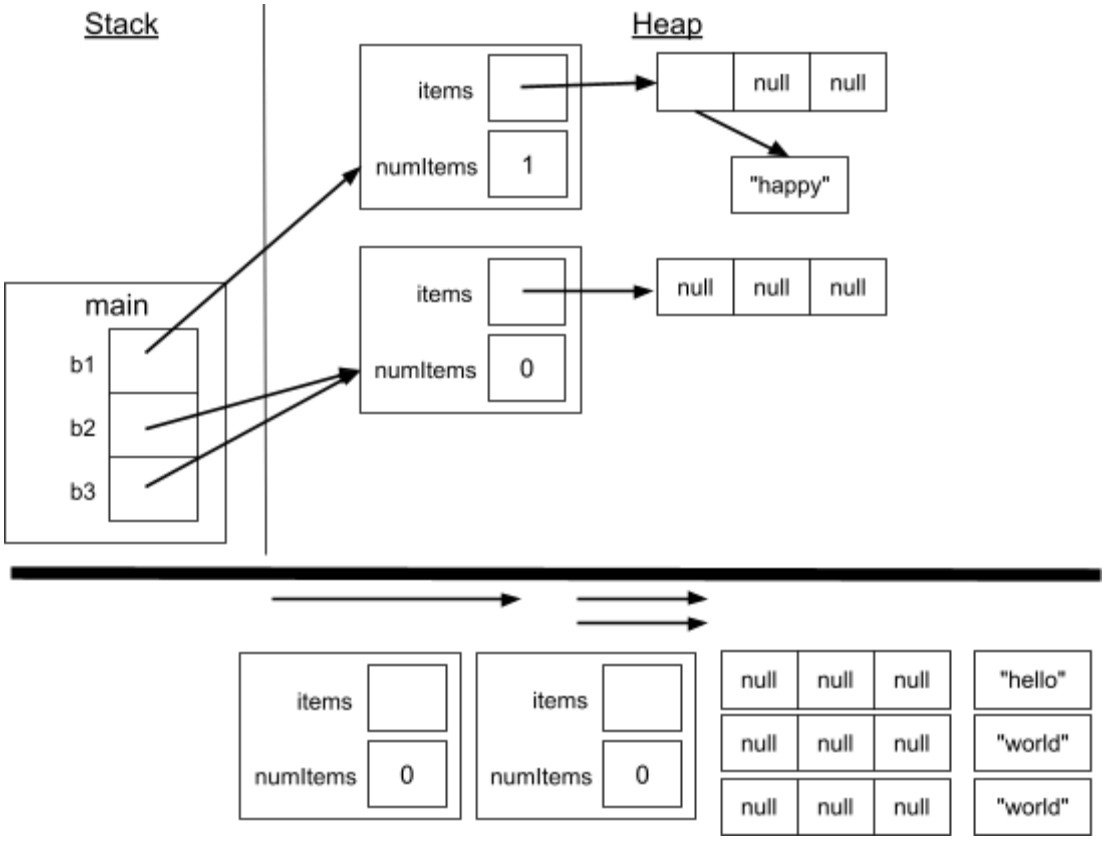
d) This is not allowed because Too is a subclass of Zoo, it only works if the object is a superclass of a subclass, not vice versa.

Problem 3: Memory management and the `ArrayBag` class

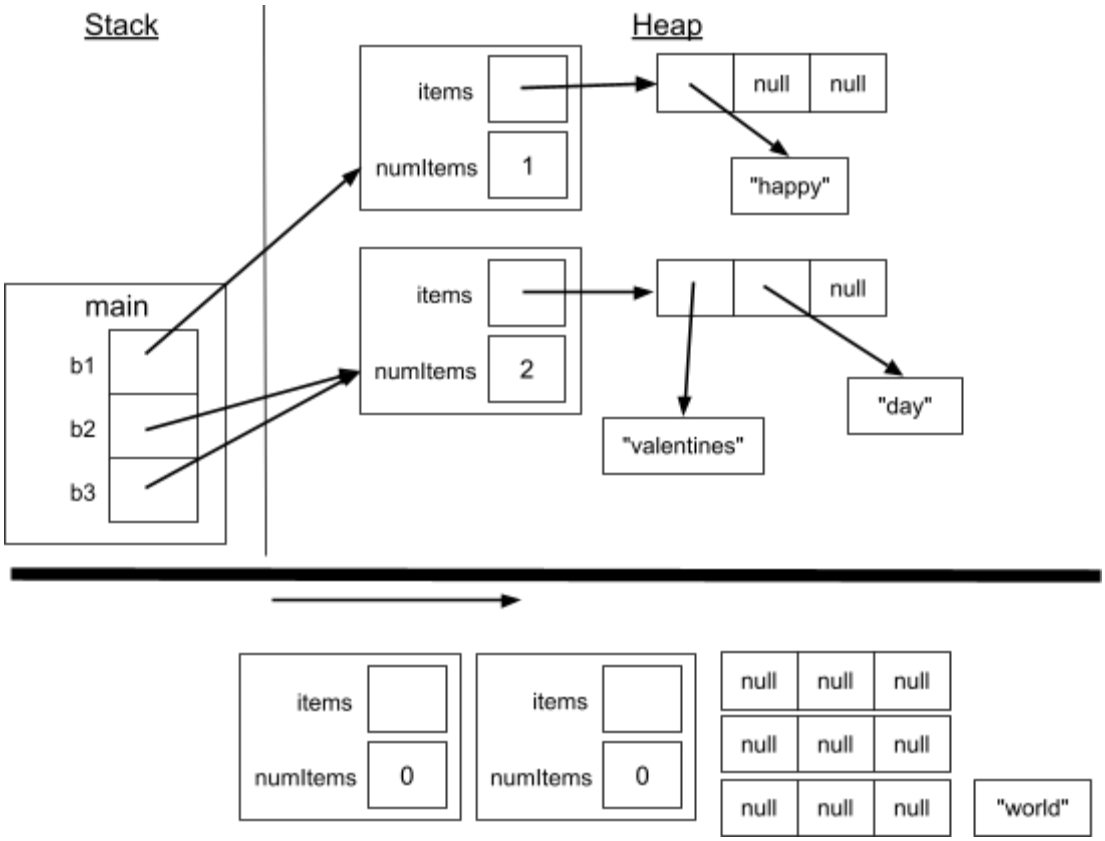
3-1)



3-2)



3-3)



3-4)

```
output:
{happy}
{valentines, day}
{valentines, day}
```