DON'T WRITE ANYTHING HERE!!!

Name: *(as it would appear on official course roster)*	Seat:
Umail address: @umail.ucsb.edu	

1

EXAM: e01: Midterm 1

e01

CS56 m18

ready?	date	points
true	Thu 08/30 09:30AM	100

You may not collaborate on this exam with anyone. If you need to use the restroom, you must leave your cell phone with the exam proctor before leaving the room.

- Write your name at the top of this page **AND EVERY ODD NUMBERED PAGE.**
- Double check that you turned in ALL pages; look for "End of Exam" on the last page.
- This exam is **closed book, closed notes**, **closed mouth, cell phone off.**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes.
- This sheet will be collected with the exam, and might not be returned.
- Please write your name on your notes sheet.

---

1. For this question, you need page 1 of handout A with code for these files: `Beverage.java`, `Edible.java`, `Food.java`, `FreeCandy.java` and `Product.java`. These are classes used by a grocery store known as “Trader Bobs”.

   Some of these methods will compile and run, while others will not.

   Indicate, for each method, whether it compiles or not, and if it does compile, the output when invoked. in context of the classes on handout A and assuming the methods appear inside this class:

   ```
   public class TraderBobs {
    // methods appear here
   }
   ```

   a. (3 pts)

   ```
   public static void TB01 () {
       Beverage b1 = new Beverage(89,"Diet Coke",0,12.0);
       System.out.println("b1: " + b1.getCalories());
   }
   ```

Will it compile?	Output when called (only if it compiles)
☐ Yes ☐ No	

   b. (3 pts)

   ```
   public static void TB02 () {
       Product p2 = new Beverage(99,"Coke",150,12.0);
       System.out.println("p2: " + p2.getName());
   }
   ```

Will it compile?	Output when called (only if it compiles)
☐ Yes ☐ No	

2

e01

CS56 m18

DON'T WRITE ANYTHING HERE!!!

2. Continued from previous problem…

Some of these methods will compile and run, while others will not.

Indicate, for each method, whether it compiles or not, and if it does compile, the output when invoked. in context of the classes on handout A and assuming the methods appear inside this class:

```
public class TraderBobs {
 // methods appear here
}
```

a. (3 pts)

```
public static void TB03 () {
    Product p3 = new Beverage(199,"Milk",130,6.75);
    System.out.println("p3: " + p3.getCalories());
}
```

Will it compile?	Output when called (only if it compiles)
☐ Yes ☐ No	

b. (3 pts)

```
public static void TB04 () {
    FreeCandy f4 = new FreeCandy(25);
    System.out.println("f4: " + f4.getCalories());
}
```

Will it compile?	Output when called (only if it compiles)
☐ Yes ☐ No	

c. (3 pts)

```
public static void TB05 () {
    Edible e5 = new FreeCandy(10);
    System.out.println("e5: " + e5.getCalories());
}
```

Will it compile?	Output when called (only if it compiles)
☐ Yes ☐ No	

d. (3 pts)

```
public static void TB06 () {
    Edible e6 = new Edible(){
            public int getCalories(){return 50;}
        };
    System.out.println("e6: " + e6.getCalories());
}
```

Will it compile?	Output when called (only if it compiles)
☐ Yes ☐ No	

DON'T WRITE ANYTHING HERE!!!

Name: *(as it would appear on official course roster)*	Seat:
Umail address: @umail.ucsb.edu	

3

e01

CS56 m18

3. Continued from previous problem…

Some of these methods will compile and run, while others will not.

Indicate, for each method, whether it compiles or not, and if it does compile, the output when invoked. in context of the classes on handout A and assuming the methods appear inside this class:

```
public class TraderBobs {
 // methods appear here
}
```

a. (3 pts)

```
public static void TB07 () {
    Food f7 = new Food(99,"Almonds",100,0.63);
    System.out.println("f7: " + f7.getName());
}
```

Will it compile?	Output when called (only if it compiles)
☐ Yes ☐ No	

b. (3 pts)

```
public static void TB08 () {
    Edible e8 = new Food(249,"Kind Bar",200,1.4);
    System.out.println("e8: " + e8.getName());
}
```

Will it compile?	Output when called (only if it compiles)
☐ Yes ☐ No	

c. (3 pts)

```
public static void TB09 () {
    Food f9 = new Food(199,"Gummi Bears",520,5);
    System.out.println("f9: " + f9.getPrice());
}
```

Will it compile?	Output when called (only if it compiles)
☐ Yes ☐ No	

d. (3 pts)

```
public static void TB10 () {
    Product p10 = new Product(299,"Ziploc Bags");
    System.out.println("p10: " + p10.getPrice());
}
```

Will it compile?	Output when called (only if it compiles)
☐ Yes ☐ No	

4

e01

CS56 m18

4. (10 pts) Refer to the code for the class `Dog` with a `main` that creates some `Dog` objects.

   Your job: figure out after which line of main() each of the following Dog objects is eligible for garbage collection.

   If an object is still not eligible for garbage collection when the last line of main is reached, write "never". Each answer should be a line number, or the word never.

DON'T WRITE ANYTHING HERE!!!

Object	Fill in line here
(a) `Fido`	
(b) `Ginger`	
(c) `Harry`	
(d) `Izzy`	
(e) `Jack`	

```
1  public class Dog {
2
3      private static Dog bestInShow = null;
4      private String name;
5
6      public static void setBestInShow(Dog b) {
7          bestInShow = b;
8      }
9
10     public static Dog getBestInShow() {
11         return bestInShow;
12     }
13
14     public Dog(String name) { this.name = name;}
15
16     public static void main(String [] args) {
17
18         Dog d1 = new Dog("Fido");
19         Dog d2 = new Dog("Ginger");
20         Dog d3 = new Dog("Harry");
21         Dog d4 = new Dog("Izzy");
22         Dog d5 = new Dog("Jack");
23         Dog d6 = d1;
24
25         setBestInShow(d3);
26         d1 = d4;
27         d3 = d6;
28         d1 = null;
29         d2 = null;
30         d3 = null;
31         d4 = null;
32         d5 = null;
33         d6 = null;
34         setBestInShow(null);
35     }
36 }
```

DON'T WRITE ANYTHING HERE!!!

Name: *(as it would appear on official course roster)*	Seat:
Umail address: @umail.ucsb.edu	

5

e01

CS56 m18

5. Given the following program below, determine whether each line of code involves boxing or unboxing, and check the boxes accordingly.

   If a line of code involves both, check both boxes. If it involves neither, check neither box.

Points	Line number	Auto-Boxing	Auto-Unboxing
(2 pts)	4	☐	☐
(2 pts)	5	☐	☐
(2 pts)	6	☐	☐
(2 pts)	7	☐	☐
(2 pts)	8	☐	☐

Points	Line number	Auto-Boxing	Auto-Unboxing
(2 pts)	9	☐	☐
(2 pts)	10	☐	☐
(2 pts)	11	☐	☐
(2 pts)	12	☐	☐

```
1  import java.util.ArrayList;
2  public class BoxUnbox {
3      public static void main(String [] args) {
4          ArrayList<Integer> mylist = new ArrayList<Integer>();
5          mylist.add(new Integer(2));
6          mylist.add(1);
7          mylist.set(0,3);
8          int a = mylist.get(0) + mylist.get(1);
9          mylist.add(7);
10         Integer b = mylist.get(2);
11         Integer c = b;
12         Integer d = a + c;
13         System.out.println("a=" + a + " b= " + b + " c=" + c + " d=" + d);
14     }
15 }
```

6. (8 pts)

   Working with the same program as in the previous problem: What is the output?

   Indicate by filling in the blanks:

a=____ b= ___ c=____ d=____

6

e01

CS56 m18

7. In lecture, we explored how a package called JaCoCo could be used to measure the test-case coverage of a piece of code. Suppose you were asked the two questions below at a job interview. How would you answer?

   a. (3 pts) If we hire you, the team you'll be working on is trying to work towards increasing the test-case coverage for the source code of company's products. I want to make sure you understand that that means. What does it mean to work towards increased test-case coverage?

   b. (6 pts) I also want to make sure you understand the importance of the task. What are some of the benefits of increased test-case coverage for it's code?

8. (5 pts) Please refer to the `Student` class on Handout B. You will see that the body of the `equals` method contains a blank. In the space below, write a line of code that can go into this blank, so that the `equals` method compares for equality based on whether the perm numbers of two students are the same.

9. (5 pts) Please refer to the `Student` class on Handout B and the `Main` class on Handout A. You will see that the parameter to the `sort` method on line 15 of `Main.java` is blank. In the space below, write an expression that could go in that blank so that the sort is by perm number.

10. (5 pts) Please refer to the `Student` class on Handout B and the `Main` class on Handout A. You will see that the parameter to the `sort` method on line 15 of `Main.java` is blank. In the space below, write an expression that could go in that blank so that the sort is by perm number.

11. (5 pts) Please refer to the `Student` class on Handout B and the `Main` class on Handout A. You see that the output for each student is formatted in a particular way when the `toString` of `Student` is implicitly invoked by the `toString` of `ArrayList<Student>`. You also see that the implementation of `toString` for Student contains a blank.

    In the space below, write a line of code that completes the `toString` method so that the return value matches what is shown for each `Student` object on Handout A.

12. (5 pts) Please refer to the `Student` class on Handout B. You see that this class implements `Comparable<Student>`, but the `compareTo` method has a blank in it.

    In the space below, write a line of code that completes the `compareTo` method so that the comparison is made in a way consistent with the javadoc comment shown on lines 37-41.

End of Exam