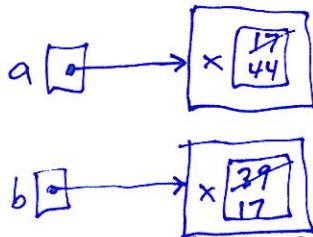


**Question 1.** [5 points] What output is printed by the following program (which begins on the left and continues on the right)?

<pre>public class Q1 {     public int x;      public Q1(int x) {         this.x = x;     } }</pre>	<pre>public static void main(String[] args) {     Q1 a = new Q1(17);     Q1 b = new Q1(39);      b.x = a.x;     a.x = 44;     System.out.printf("%d,%d\n", a.x, b.x); } }</pre>
--	---



output:  
44 , 17

**Question 2.** [5 points] What output is printed by the following program?

```
public class Q2 {
    public static void main(String[] args) {
        int[] a = new int[1];
        int[] b;

        a[0] = 99;
        b = a; // make b refer to the same array as a
        b[0] = 101;
        System.out.printf("%d,%d\n", a[0], b[0]);
    }
}
```

output :  
101 , 101

**Question 3.** [10 points] Complete the following method. It should return the average of the minimum and maximum of the values stored in the `values` array. You may assume that the array will have at least one element.

```

public static double avgMinAndMax(double[] values) {
    double min = values [0];
    double max = values [1];

    for (int i = 1; i < values.length; i++) {
        if (values [i] < min) {
            min = values [i];
        }
        if (values [i] > max) {
            max = values [i];
        }
    }

    return (min + max) / 2;
}

```

**Question 4.** [5 points] What output is printed by the following program (which begins on the left and continues on the right)?

<pre> public class Q4 {     public static void f(int x)         throws Exception {          if (x % 2 == 0) {             throw new Exception();         }         System.out.printf(             "f(%d)\n", x);     } } </pre>	<pre> public static void main(String[] a) {     int[] values = { 1, 2, 3, 4 };     for (int v : values) {         try {             f(v);         } catch (Exception e) {             System.out.printf("Exception!\n");         }     } } } </pre>
---	---

f(1),  
 Exception!  
 f(3)  
 Exception!

**Question 5.** [10 points] Consider the following program:

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
import java.util.Scanner;

public class Q5 {
    public static void main(String[] a) throws IOException {
        Scanner keyboard = new Scanner(System.in);
        System.out.print("What file? ");
        String fileName = keyboard.nextLine();

        BufferedReader r = new BufferedReader(new FileReader(fileName));
        while (r.readLine() != null) {
            System.out.println(r.readLine());
        }

        r.close();
    }
}
```

(a) In words, explain what happens when this program is executed. Assume that the user types the name of a text file that does exist and is readable.

It prints every other line of text from the file, starting with the second line.

(b) Is the program guaranteed to close the `BufferedReader` if the file is opened successfully? Why or why not? Explain briefly.

No, because an `IOException` could be thrown out of the method from either of the calls to `readLine()`, resulting in the call to `r.close()` not being executed.

**Question 6.** [10 points] Consider the following JUnit test class (which begins on the left and continues on the right):

<pre>public class ComboLockTest {     private ComboLock lock;      @Before     public void setUp() {         lock = new ComboLock(21, 8, 14);     } }</pre>	<pre>@Test public void testInvalidCombo() {     lock.spin(21);     lock.spin(9);     lock.spin(14);     assertFalse(lock.isUnlocked()); }  @Test public void testValidCombo() {     lock.spin(21);     lock.spin(8);     lock.spin(14);     assertTrue(lock.isUnlocked()); } }</pre>
---	--

Show how the `ComboLock` class would be defined. You should show all of the public methods, but you do **not** need to show how the methods would be implemented. (Just leave the body of each method empty.)

```
public class ComboLock {
    // Fields ...

    public ComboLock(int a, int b, int c) {
        :
    }

    public void spin(int x) {
        :
    }

    public boolean isUnlocked() {
        :
    }
}
```

Question 7. [15 points] Consider the following Animal class and partially-specified FruitBat class:

<pre>public abstract class Animal {     private String sound;      public Animal(String s) {         this.sound = s;     }      public String getSound() {         return sound;     }      public abstract     boolean eatsInsects(); }</pre>	<pre>public class FruitBat <span style="border: 1px solid black; padding: 2px;">Missing code 1</span> {     public FruitBat() {         <span style="border: 1px solid black; padding: 2px;">Missing code 2</span>     }      <span style="border: 1px solid black; padding: 2px;">Missing code 3</span> }</pre>
--	--

Now, consider the following statements:

```
Animal fruitBat = new FruitBat();
System.out.println(fruitBat.getSound());
System.out.println(fruitBat.eatsInsects());
```

Specify code that can be substituted for each of the three Missing code blocks that will allow these statements to compile and run successfully and print the output

Squeak  
true

Use the other side of the page if necessary.

Missing code 1  
extends Animal

Missing code 2  
super("Squeak");

Missing code 3  
public boolean eatsInsects() {  
 return true;  
}