

1) (30 points) Write the output of each piece of code. If the code gives an error, write any output that would happen before the error, and then write "ERROR".

	Code	Output
a) (5 points)	<pre>int a = 11; int b = 2; double c = 5; System.out.println(((double) a)/b); System.out.println(c/b); System.out.println(b/a); System.out.println(a/b);</pre>	<pre>5.5 2.5 0 5</pre>
b) (5 points)	<pre>String k = "ketchup"; String m = "mustard"; System.out.println(k.substring(5) + m.substring(0,4) + k.charAt(3));</pre>	<pre>upmustc</pre>
c) (5 points)	<pre>for(int i = 3; i < 9; i++) { System.out.println(i%2); }</pre>	<pre>1 0 1 0 1 0</pre>
d) (5 points)	<pre>String[] a = {"aardvark", "ocelot", "capybara"}; System.out.println(a[2]); String[] b = a; System.out.println(b[2]); b[2] = "echidna"; System.out.println(a[2]); System.out.println(b[0]); System.out.println(a[3]);</pre>	<pre>capybara capybara echidna aardvark ERROR</pre>

1) (Continued)

	Code	Output
<p>e) (5 points)</p>	<pre>String w = "whit"; int k = 0; int len = w.length(); boolean keepGoing = true; while (keepGoing) { System.out.println(w.substring(k, len)); k++; if (k == len) { keepGoing = false; } }</pre>	<pre>whit hit it t</pre>
<p>f) (5 points)</p>	<pre>int sum = 0, j = 2; while(sum < 60) { switch(j) { case 0: j = 2; sum -= 10; break; case 1: j = 0; sum += 10; case 2: j = 1; sum += 20; break; } System.out.println(sum); }</pre>	<pre>20 50 80</pre>

2) (30 points) The Java class called `CUNYCollege` is started below. An instance of class `CUNYCollege` represents one of the CUNY colleges. This class has four private instance variables:

- `name`, which is a `String` representing the name of the college.
- `numStudents`, which is an `int` representing the number of students at the college.
- `numStudentsCS`, which is an `int` representing the number of students at the college in the computer science program.
- `isSenior`, which is a `boolean` representing whether the CUNY college is a senior one (like Lehman) or not

```
public class CUNYCollege {
    private String name;
    private int numStudents;
    private int numStudentsCS;
    private boolean isSenior;

    // your code will go here

}
```

a) (7 points) Write a constructor which takes in a name, number of students at the college, number of students in computer science, and boolean of whether the college is a senior college or not, and assigns those values to a `CUNYCollege` object.

```
public CUNYCollege(String n, int numS, int numCS, boolean isSenior) {
    name = n;
    numStudents = numS;
    numStudentsCS = numCS;
    this.isSenior = isSenior;
}
```

b) (5 points) Use your constructor from part (a) to instantiate an object of class `CUNYCollege` with name "Lehman", 10,326 students in the college, 150 students in computer science, and a senior college.

```
CUNYCollege lehman = new CUNYCollege("Lehman",10326,150,true);
```

2) (continued)

c) (8 points) Write a non-static method `equals(CUNYCollege otherCollege)`, which compares two `CUNYColleges` and returns `true` if the number of computer science students and the type of college (senior or non-senior) are the same for both `CUNYColleges`, and `false` otherwise.

```
//example overriding inherited equals
@Override
public boolean equals(Object otherObj){
    if(otherObj instanceof CUNYCollege){
        CUNYCollege otherCollege = (CUNYCollege)otherObj;
        return ((numStudentsCS == otherCollege.numStudentsCS) &&
                (isSenior == otherCollege.isSenior));
    }
    return false;
}

//example without @Override
public boolean equals(CUNYCollege otherCollege) {
    return ((numStudentsCS == otherCollege.numStudentsCS) &&
            (isSenior == otherCollege.isSenior));
}
```

d) (10 points) Write a static method `listOfColleges`, which takes in an array of `CUNYCollege` objects, and returns a single `String` containing the names of the colleges in the array, separated by spaces. Assume that every element of the array contains a `CUNYCollege` instance.

```
public static String listOfColleges(CUNYCollege[] array) {
    String s = "";
    for (int i = 0; i < array.length; i++) {
        s = s + array[i].name + " ";
    }
    return s;
}
```

3) (10 points) Write a static method called `zeroOut` that accepts two input parameters:

- an `int` called `num`
- an `int` array called `array`

This method should change all of the elements of `array` that are less than or equal to `num` to be 0. The method should not return anything.

For example, the array

4	-9	9	57	2
---	----	---	----	---

would become

0	0	9	57	0
---	---	---	----	---

when calling `zeroOut` with `num = 4`.

```
public static void zeroOut(int num, int[] array) {
    for(int i =0; i < array.length; i++) {
        if (array[i] <= num) {
            array[i] = 0;
        }
    }
}
```

4) (20 points) The Java class `Borough` (code below) represents a borough.

a) (6 points) Fill in the code below so that it has the following private attributes:

- the name of the borough (`name`)
- area of the borough in square miles (`area`)
- whether the borough is located on an island (`isOnIsland`)

```
public class Borough{
    //your code goes here

    private String name;
    private double area;
    private boolean isOnIsland;

    public Borough(){
        name = "no name yet";
        area = 0.0;
        isOnIsland = true;
    }
}
```

b) (4 points) Write the accessor (getter) method for the instance variable `area`.

```
public double getArea() {
    return area;
}
```

c) (5 points) Write the mutator (setter) method for the instance variable `area`.
If the input argument is negative, do not change the variable's value.

```
public void setArea(double area) {
    if (area >= 0) {
        this.area = area;
    }
}
//without this.
public void setArea(double a) {
    if (a >= 0) {
        area = a;
    }
}
```

- d) (5 points) Write a `toString` method for the class `Borough` that returns a `String` containing the name, the area, and the `isOnIsland` of the `Borough` Object. Format the returned `String` as shown in the example below:

Name: The Boogie Down, Area: 57.43 sq. mi., not on an island

Name: Brooklyn, Area: 96.91 sq. mi., on an island

```
public String toString() {
    String s = "Name: " + name + ", Area: " + area + "sq. mi., ";
    if (isOnIsland) {
        s = s + "on an island";
    }
    else {
        s = s + " not on an island";
    }
    return s;
}
```

5) (10 points) Write a piece of code that repeatedly asks the user "Who's there?", and allows them to enter a word. The code should repeat this until the user enters the word "Orange", in which case the code should print out "Orange you glad you didn't say Banana!" and exit. If at any time the user enters the word "Banana", then the code should print out "Banana Who?!", before going back to asking the user "Who's there?"

```
public static void question5_dowhile(){
    Scanner keyboard = new Scanner(System.in);
    System.out.println("Who \'s there?");
    String reply = "";
    do {
        reply = keyboard.nextLine();
        if (reply.equals("Banana")) {
            System.out.println("Banana Who?!");
        }else if (!(reply.equals("Orange"))){
            System.out.println("Who \'s there?");
        }
    } while (!(reply.equals("Orange")));
    keyboard.close();
    System.out.println("Orange you glad you didn't say Banana!");
}
```

```
public static void question5_while(){
    Scanner keyboard = new Scanner(System.in);
    System.out.println("Who \'s there?");
    String reply = "";
    reply = keyboard.nextLine();
    while (!(reply.equals("Orange"))){
        if (reply.equals("Banana")) {
            System.out.println("Banana Who?!");
        }else{
            System.out.println("Who \'s there?");
        }
        reply = keyboard.nextLine();
    }
    keyboard.close();
    System.out.println("Orange you glad you didn't say Banana!");
}
```