Name _____ Period_____

1. The `LightBoard` class models a two-dimensional display of lights, where each light is either on or off, as represented by a Boolean value. You will implement a constructor to initialize the display and a method to evaluate a light.

```
public class LightBoard
{
     /** The lights on the board, where true represents on and false
      *  represents off.
      */

     private boolean[][] lights;

     /** Constructs a LightBoard object having numRows rows and numCols columns
      *  Precondition: numRows > 0, numCols > 0
      *  Postcondition: each light has a 60% probability of being set to on
      */

     public LightBoard(int numRows, int numCols)
     {     /* To be implemented in part (a)  */    }

     /** Evaluates a light in row index row and column index col
      *  and returns a status as described in part (b).
      *  Precondition: row and col are valid indexes in lights.
      */

     public boolean evaluateLight(int row, int col)
     { /* to be implemented in part (b) */ }

      // There may be additional instance variables, constructors, and methods not
shown.
}
```

(a) Write the constructor for the `LightBoard` class, which initializes lights so that each light is set to on with a 60% probability. The notation `lights[r][c]` represents the array element at row `r` and column `c`.

Complete the `LightBoard` constructor below.

```
/** Constructs a LightBoard object having numRows rows and numCols columns.
 * Precondition: numRows > 0, numCols > 0
 * Postcondition: each light has a 60% probability of being set to on.
 */

public LightBoard(int numRows, int numCols) {



       lights = new boolean[numRows][numCols];
        for(int row = 0; row < lights.length; row++){
            for(int col = 0; col < lights[row].length; col++){
                lights[row][col] = (Math.random() < .6);
            }
        }
```

```
}
```

/4

Score _____/21

(b) Write the method `evaluateLight`, which computes and returns the status of a light at a given row and column based on the following rules.

1. If the light is off, return false if the number of lights in its column that are on is odd, including the current light.
2. If the light is on, return true if the number of lights in its column that are on is divisible by three.
3. Otherwise, return the light's current status.

Class information for this question

```
public class LightBoard
private boolean[][] lights
public LightBoard(int numRows, int numCols)
public boolean evaluateLight(int row, int col)
```

Complete the evaluateLight method below.

```
/** Evaluates a light in row index row and column index col and returns a status
 * as described in part (b).
 * Precondition: row and col are valid indexes in lights.
 */


public boolean evaluateLight(int row, int col) {



        int count = 0;
        for(int r = 0; r < lights.length; r++){
            if(lights[r][col]){
                count++;
            }
        }
        if(!lights[row][col] && count%2!=0){
            return false;
        }
        if(lights[row][col] && count%3 == 0){
            return true;
        }
        return lights[row][col];









}
```

/5

2. Refer to the code below,

```
public class Pet {

    private boolean vegetarian;
    private String type;
    private final int noOfLegs;

    public Pet(boolean vegetarian, String type, int noOfLegs){
        //sets the variables vegetarian, type, and noOfLegs declared
        //in this class to the parameters passed in the constructor
        this.vegetarian = veg;
        this.type = type;
        this.noOfLegs = legs;
    }

    public boolean getEats(){
        return vegetarian;
    }

    public int getLegs(){
        return noOfLegs;
    }

    Public int getType(){
        return type;
    }

    public String toString() {
                return "I am a : " + type;
    }
}

public class Cat extends Pet{

    public String name;

    public Cat(String name){
        super(false, "cat", 4);
        this.name = name;
    }

    public void speak(){
        System.out.println("Meow!");
    }

    public String toString() {
        return super.toString() + "\nMy name is : " + name;
    }
}
```

```
public class Fish extends Pet{

    public String name;
    public Fish(String n){
        super(false, "fish", 0);
        this.name = n;
    }

    public void speak(){
        System.out.println("Blub, Blub");
    }

    public String toString() {
                return "My name is " + name;
    }

}
```

(a) What is/are the parent class(es) associated with the Fish class?

Pet, Object

/1

(b) For each of the following (i) Indicate whether the statement is valid (V) or invalid (I) (ii) If the statement is not valid, indicate why.

Statement	V/I	If "I", indicate why.
`Fish f = new Fish(false, "Fish", 0);`	**I**	**Wrong parameters**
`Cat c = new Fish("Fred");`	**I**	**A Fish is not a Cat**
`Fish f = new Pet(true, "Fish", 0);`	**I**	**A Pet is not a Fish**
`Pet p = new Fish("Dori");`	**V**	
`Object o = new Cat("Fred");`	**V**	
`Object o = new Pet(317, 555, 1000);`	**I**	**Wrong parameters**

/6

(c) Refer to the code block below to indicate what is printed for each of the following statements. If an error occurs write "ERROR" AND indicate why the error occurs.

```
Pet p = new Pet(true, "Spider", 8);
Cat c = new Cat("Roscoe");
Fish f = new Fish("Nemo");
```

(i) `System.out.println(new Fish("Bob") instanceof Pet);`

True

Score _____/21

(ii)
```
System.out.println(p);
```

I am a: Spider

(iii)
```
System.out.println(c);
```

I am a: cat
My name is Roscoe

(iv)
```
Pet[] myPets = new Pet[2];
myPets[0] = p;
myPets[1] = c;
myPets[0].speak();
```

Error; speak() is not in the Pet class

(v)
```
f.speak();
```

Blub, Blub

(vi)
```
System.out.println(f.getLegs());
```

0

(vii)
```
System.out.println(c.getEats());
```

false

(viii)
```
System.out.println(p.toString());
```

I am a: Spider

(ix)
```
System.out.println(new Cat() instanceof Pet);
```

Error; Cat requires a parameter

/9

Score _____/21

3. Refer to the code below,

```
class A {
    public A() {
        System.out.println("Inside A's constructor");
    }
}
class B extends A {
    public B() {
        System.out.println("Inside B's constructor");
    }
}
class C extends B {
    public C() {
        System.out.println("Inside C's constructor");
    }
}
public class Inheritance {
    public static void main(String[] args) {
        /** Statements for questions go here **/
    }
}
```

(a) After executing the statement `A b = new C();`, what is output by the program?

Inside A's constructor
Inside B's constructor
Inside C's constructor

***NOTE: :The constructor of the super class is executed BEFORE the subclass**

/1

(b) After executing the statement `B a = new B();`, what is ouptut by the program?

Inside A's constructor
Inside B's constructor

***Because B extends A, A's constructor is executed first**

/1

(c) What is the output of the following statement? System.out.println((new A()) instanceof A);

Inside A's constructor
true

/1

(d) What is the output of the following statement? System.out.println((new A() instanceof B);

Inside A's constructor
false

/1

Score _____/21

(e) What is the output of the following statement? System.out.println((new C() instanceof B);

Inside A's constructor
Inside B's constructor
Inside C's constructor
true

/1

Score _____/21