Question 1 [20 marks]

In the code on the right, ***clearly circle and identify by number*** one (and only one) example of each of the items in the list on the left. If you believe no example exists in the given code, write *"no example"* next to that item.

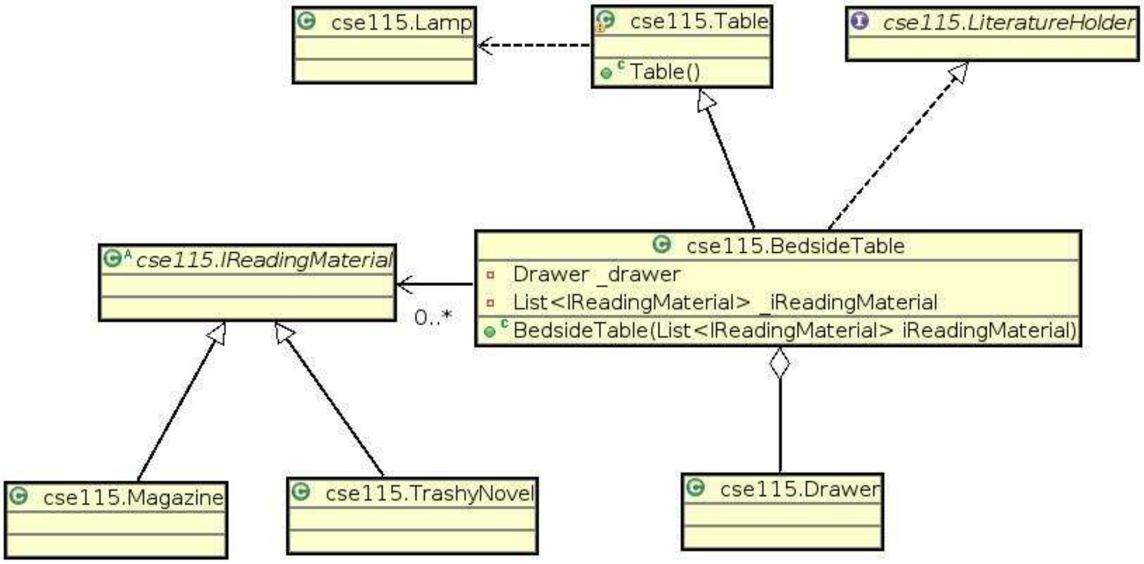
1. *constructor definition*
2. *parameter (formal parameter) declaration*
3. *method call*
4. *argument (actual parameter) in method call*
5. *method return type specification*
6. *instance variable declaration*
7. *assignment statement*
8. *accessor method*
9. *mutator method*
10. *method overriding*
11. *method overloading*
12. *name of a subclass*
13. *name of a superclass*
14. *abstract method declaration*
15. *explicit call to a superclass' constructor*
16. *a loop*
17. *the name of a primitive data type*
18. *a boolean expression*
19. *package name*
20. *name of an interface*

```
package MysteryShapes;

public class MysteryBall
       extends MysteryShapes.Shape
       implements MysteryShapes.IBall
{
   private Behavior _b;
   private String _name;
   private BehaviorHolder _holder;
   private int _counter;

   public MysteryBall(DrawingPanel dp, String n,
                      BehaviorHolder h)
   {
       super(dp);
       this.setBehavior(h.getBehavior());
       _name = n;
       _holder = h;
       _counter = 0;
   }
   public MysteryBall(DrawingPanel dp, BehaviorHolder h)
   {
       super(dp);
       this.setBehavior(h.getBehavior());
       _name = "<None assigned>";
       _holder = h;
       _counter = 0;
   }
   public void setBehavior(Behavior b)
   {
       _b = b;
   }
   public Behavior getBehavior()
   {
       return _b;
   }
   public void update()
   {
       super.update();
       _b.perform(this);
   }
   public void react()
   {
       _counter = _counter + 1;
       if (_counter == 3) {
           this.setBehavior(_holder.getBehavior());
           _counter = 0;
}  }  }
```

Question 2

Part a **[5 marks]**

Given the following UML diagram, give one (and *only* one) example of two classes involved in the each relationship listed, as well as the direction of the relationship (i.e. *R is-a K* is different from *K is-a R* . If the relationship does not exist in the diagram, write NONE.



- Composition (*has-a*)

- Association (*knows-a*)

- Dependency (*uses-a*)

- Inheritance (*is-a*)

- Implements (*realization*)

Question 2, part b [15 marks]

Write the code for class BedsideTable that would be generated if you were to draw the diagram from part (a) using Green.

Question 3 [10 marks]

Consider the method definition below.

```
public void foo(int x) {
    if (x > 5) {
        System.out.println("Passing");
    }
    else if (x > 10) {
        System.out.println("Doing well");
    }
    else if (x > 20) {
        System.out.println("Exceptional work!");
    }
    else {
        System.out.println("Failing");
    }
}
```

State what will be printed as a result of each of the following method calls:

```
foo(3);
```

```
foo(10);
```

```
foo(7);
```

```
foo(15);
```

```
foo(25);
```

Question 4 [10 marks]

In class we discussed the game of tic-tac-toe, which is played on a 3 by 3 game board. Suppose that the game were extended to be a game of seven-in-a-row played on a 100 by 100 game board.

Suppose further that your job is to initialize the board with `Cell` objects. In other words, you need to write code which will initialize each position on the board with a new `Cell` object. Assume that the `Cell` class defines a constructor which takes a `Position` object indicating the row and column of the board position which the `Cell` object will occupy.

Hint: use loops!

Question 5 [10 marks]

Consider the following modification to *Number Muncher*, which is a timed version of the game. The way it works is this: the player has a set amount of time to clear a level. Rather than immediately removing a number from the board when it is munched, the game makes a note of the munched numbers by storing references to them in a suitable collection, such as `ArrayList<Number>`. When a timer expires, the collection is traversed and all the numbers stored in are removed from the list.

Assume the collection `toBeDeleted` holds the collection of `Number` objects to be removed. Fill in the `delete()` method so that it iterates over each number in `toBeDeleted` and calls the `deleteNumber()` method on each `Number`; assume the `Number` class has already been written for you, and that the `deleteNumber()` method is defined in it.

```
public void delete(java.util.ArrayList<Number> toBeDeleted) {









}
```

Question 6 [10 marks]

You have been hired by the 3N company (formerly known as *NiftyNikNaks*). Your job is to write part of the code for a universal remote-control.

The remote control doesn't do much just yet: it can turn things on and it can turn them off. The nifty thing about it is that in order to do this all the remote needs to do is communicate with the device through a *driver* (an object which can interface with the hardware of the device being controlled). A driver must implement the `IDriver` interface:

```
public interface IDriver {
    public void turnOn();
    public void turnOff();
}
```

Define the `onButtonPressed` and `offButtonPressed` methods so that when the `onButtonPressed` method is called, the currently selected device is turned on, and similarly when the `offButtonPressed` method is called, the currently selected device is turned off.

```
public class UniversalRemote {

    private HashMap<String,IDriver> _devices;    // map of known devices
    private String _nameOfSelectedDevice;        // name of currently selected device

    public UnivesalRemote() {
        _devices = new HashMap<String,IDriver>();
    }

    public void addDevice(String name, IDriver d) {
        _devices.put(name, d);
    }

    public void selectDevice(String name) {
        _nameOfSelectedDevice = name;
    }

    public void onButtonPressed() {








    }

    public void offButtonPressed() {








    }
}
```

Question 7

In class you worked in small groups on the problem of modelling the domain of the tic-tac-toe game. This question asks you to model (on your own!) a system to water house plants. The scenario, modified from a description found at `www.ext.nodak.edu/county/cass/horticulture/inform/indoor/water.htm`, is as follows:

> Watering Houseplants
>
> "How often should I water my houseplants?" There is no absolute answer to this question. It is impossible to set a time schedule for watering plants. The frequency of watering depends on many factors such as type and size of plant, temperature and humidity within the home, weather conditions and season of the year.
>
> In general, it is a sound practice for most house plants to let the soil dry out somewhat between watering ; soil moisture indicators can be purchased. These are very useful in getting to know the water needs of your plants. Of course, there are exceptions to the above general rule. Plants such as the Boston Fern, Caladium and Zebra plant won't tolerate dry soil. They should be watered often enough to keep the soil surface from becoming dry. Dieffenbachia and Schefflera plants need less water than the normal plant. They should be watered until the soil has been dry to the touch for one or two days. Cactus and Succulents prefer a dry environment and thus the soil should be allowed to dry out well between watering.
>
> The next question to answer is: "How much water should be applied?" The general rule is to add enough water so that a small amount runs out of the drain hole in the bottom of the pot. This guarantees that all the roots get properly watered. It also washes some of the excess salts (fertilizer residues) out of the pot.
>
> When watering your plants, the water temperature is particularly significant. Don't use ice cold water, especially on tropical plants. The water should be at room temperature and applied in the morning whenever possible. Allowing tap water to warm up to room temperature also allows water additives to evaporate or settle out in the water. These additives can cause the browning of plant leaf tips. The prolonged use of water from a softener usually results in poor plant growth.

Part a [10 marks]

Write down a list of candidate classes (at least 5), drawn from the description given, that you would consider helpful in implementing an automatic houseplant watering control system.

For one (and only one!) of your classes, describe its responsibilities in terms of the messages it can respond to.

Question 7, part b **[10 marks]**

The above description contains a fairly obvious reference to one of the design patterns we have studied this semester. Name the design pattern, and explain how it manifests itself in the description.

CSE 115 **Introduction to Computer Science I** **Fall 2005**

Final examination — 20% of final grade

Name (PRINT): ______________________

December 13, 2005 [Exam version 2]

Signature: ______________________

180 minutes Exam # ____________

Person #: ______________________

This examination has 10 pages. Check that you have a complete paper.

Each candidate should be prepared to produce, upon request, his or her SUNY/UB card.

This examination has 7 questions. Answer all 7 questions.

READ AND OBSERVE THE FOLLOWING RULES:

▷ Write the following, in ink, in the space provided above,

- ◇ your name, surname first,
- ◇ your usual signature,
- ◇ your UB person number, and
- ◇ your lab/recitation section.

▷ All of your writing must be handed in. This booklet must not be torn or mutilated in any way, and must not be taken from the examination room.

▷ Show all of your work in arriving at an answer, unless instructed otherwise. Partial credit will be awarded as appropriate.

▷ No candidate shall be permitted to enter the examination room after the expiration of one-half hour, or to leave during the first half-hour of the examination.

▷ No candidate shall be permitted to leave the examination room during the last fifteen minutes of the examination.

▷ Candidates are not permitted to ask questions of the invigilators, except in cases of supposed errors or ambiguities in examination questions.

▷ **CAUTION** — Candidates guilty of any of the following, or similar, dishonest practices shall be immediately dismissed from the examination and shall be liable to disciplinary action.

- ◇ Making use of any books, papers or memoranda, calculators or computers, audio or visual cassette players, or other memory aid devices, other than those authorised by the examiners.
- ◇ Speaking or communicating with other candidates.
- ◇ Purposely exposing written papers to the view of other candidates. The plea of accident or forgetfulness shall not be received.

Do not write below this line!!

Question	Points
1	/20
2	/20
3	/10
4	/10
5	/10
6	/10
7	/20
TOTAL	/100