1. The First Problem

You are supposed to answer these questions without programming them.

```
public class Alpha {
    private int a = 0;
    public void alpha() {
        a++;
        System.out.println(a);
    }
    public static void main (String [] args) {
        Alpha a = new Alpha();
        a.alpha();
    }
}
```

Output from executing the Alpha class:

```
public class Beta extends Alpha {
    protected int a = 0;
    public Beta(int x) {
        this.a = x;
    }
    public void beta() {
        a+=2;
        System.out.println(a);
    }
    public static void main (String [] args) {
        Alpha a = new Alpha();
        a.alpha();
        Beta b = new Beta(2);
        b.alpha();
        b.beta();
    }
}
```

Output from executing the Beta class:

```
public class Nu extends Beta {
    public Nu() {
        super(10);
    }
    public void nu() {
        a+=5;
        System.out.println(a);
    }
```

```
    public static void main (String [] args) {
        Nu n = new Nu();
        n.nu();
        n.beta();
        n.alpha();
        Beta b = new Beta(10);
        b.alpha();
        b.beta();
    }
}
```

Output from executing the Nu class:

2. The Second Problem

Programmers of C will understand how type checking in C was all done by the compiler at compile time. When C' s type checking system proved to restrictive for the problem at hand, the programmer was forced to bypass the type system using unsafe programming techniques such as casts, unions and pointers to void.

Java' s type checking system is more powerful than C so it is no longer necessary nor possible to use unsafe practices like in C. Java has an extensive system of type checking that is done at compile time, as well as a run-time type system that C lacks. This exercise will show you how to access the run-time type information provided by the Java language.

Questions

1. Fetch the file ShapeTest.java (student-files.zip) which contains four classes: Shape, Circle, Triangle and a tester class ShapeTest.
2. By copying the pattern for the Circle and Triangle classes, write a class Rectangle that inherits from Shape, has two properties width and length, and a two-argument constructor that sets these properties.
3. In the doStuff method, add a new Rectangle(…) to the end of the myShape array by coping the pattern for the other shapes.
4. Complete the printNames method which scans along the myShapes array to dermine at run time, what kind of shape is in each slot of the myShapes array.
5. Complete the printAreas method which scans along the myShapes array to print out the area of each shape.

3. The Third Problem: Modeling the Gourmet Milk System

Prerequisites, Goal, and Outcomes

Prerequisites: Before you begin this lab assignment, you need mastery of the following:

- UML (Knowledge of class diagram notation)
- Object-Oriented Design (Knowledge of modeling concepts: 1) Identifying classes; 2)

Identifying relationships between classes; 3) Identifying class attributes; 4) Identifying class methods

Goals: Reinforce your object-oriented design skills

Outcomes: You will master the following skills:

- Produce a UML class diagram, from a specification, that shows: 1) classes; 2) attributes; 3) methods; 4) relationships

Background

This lab assignment asks you to model a milk store application.

Description

Gourmet Milk is a store that sells milk from countries around the globe. It also sells milk brewing machines and other accessories for milk consumption. The Gourmet Milk System maintains a product catalog, processes orders, and tracks the store' s sales.

The catalog maintains the following information about the store' s products:

- Milk
  - Code
  - Description
  - Price
  - Country of origin
  - Type of heat
  - Flavor
  - Aroma
  - Acidity
  - Body
- Milk brewer
  - Code
  - Description
  - Price
  - Model of the brewer
  - Type of the water supply: Pour-over or Automatic
  - Capacity: number of cups
- Milk accessory
  - Code
  - Description
  - Price

The Gourmet Milk System processes orders. An order contains a list of products, their quantities, and the total cost. The following is an example of an order:

Quantity	Code	Description	Price
2	A001	Almond Flavored Syrup	9.00
1	C001	Colombia, Whole, 1	17.99

		lb	
1	B001	Home Milk Brewer	150.00
Order Total:	185.99		

In the Gourmet Milk System, the user can:

- Display the catalog: lists the code and description of each product
- Display a product
- Display the current order: list quantity, code, and price of each product in the current order, and the total of the order
- Add a product to the current order --- if the specified product is already part of the order, this command will modify the quantity of that product
- Remove a product from the current order
- Register the sale of the current order --- this command adds the order to the store's sales and empties the current order
- Display the sales: lists all the orders that have been sold

Tasks

These steps will guide you for completing this assignment:

1. Identifying the following:
   - The classes
   - The association relationships (include direction, multiplicity, and association attribute)
   - The specialization/generalization relationships
   - The attributes of each class
   - The methods of each class

   Your class diagram should include:
   - The class of the gourmet milk application
   - The accessor methods
   - The mutator methods if are needed
   - For the collections:
     - The methods to add and access elements
     - The methods to remove elements if are needed
   - The methods that compute other values not included in the attributes.

   Use Sun's coding conventions when naming classes, methods, and attributes.
2. Use Eclipse, Violet, PowerPoint, or another tool of your choosing to draw a UML class diagram.
3. Save the UML class diagram in a SVG, GIF, or JPG format in a file named uml-gou-milk.