How would you write a program to solve this problem?

Which one is the better deal?

• $4.99

- diameter (4 inches by 6 inches)

- rectangular (5 inches by 6 inches)

The special comes in two shapes:

- Pizza Mania is having a special on the same kind of pizza (i.e. same thickness and same ingredients).

The best little pizzeria in Texas

Pizza Mania:

January 19, 2001
What is a "better deal"?
- The Pizza’s shape is what we call a “variant”.

- The algorithm to compute the area depends on the shape of the Pizza.

- I designed a Pizza object that can tell me its price and its area.

Objects are the only things that can perform computations.

OOP Principle No. 0
How does a pizza object compute its area?

- This is an example of what we call the union pattern.

- Classes Circle and Rectangle are concrete variants of AbstractShape.

  Shapes, each with its own way of computing its area.

  Create AbstractShape class, to represent the union of all possible

Encapsulate all related variants into an "abstract class".

OOP Principle No. 1
- The Pizza object will forward all requests to compute its area to its
  reference to the given Shape instance.

- For this reason, Shape is said to be polymorphic.

- A Pizza object maintains a reference to an abstract Shape object
  which, at run time, should be an instance of a specific concrete subclass.

Program to the (abstract) interface.

OOP Principle No. 2
This is a design pattern called the Strategy pattern.

of reducing code complexity.

This has the effect
shape for the Pizza object to compute its area. This has the effect
– There is no conditional statement to check for the specific type of

and the concrete subclasses of Shape.

This has the effect of reducing the "compiling" between the Pizza class
not care about) what kind of concrete shape its Shape reference is. It is important to realize that the Pizza object does not know (and does
Strategy pattern consists of a union pattern of strategies.

In general, the Strategy pattern consists of a union pattern of strategies.

The Strategy pattern plays the role of the (abstract)

Pizza class, and the abstract Asshape interface in the union. In our Pizza example, the context is the

abstract Strategy in the union. In our Pizza example, the context is the

and a client class, called the context, that contains a reference to the

In general, the Strategy pattern consists of a union pattern of strategies.
"messages" (by calling their methods) to perform the desired tasks.

The main method will instantiate appropriate objects and send them

```java
public static void main(String[] args) {
    // signature
    One of them must be public and must have a method with the following

    // A Java program consists of one or more classes.

    Java Programs
}
```