

Reference

Material for these slides is taken from many UML reference books. However, the two I most often used are:

**UML Explained, by Kendall Scott, © 2001
Addison-Wesley, ISBN: 0-201-72182-1 (~\$30)**

Teach Yourself UML in 24 Hours, by Joseph Schmuller, © 2002 Sams Publishing, ISBN: 0-672-32238-2 (~\$30)

But, there are lots of others. (A quick count of my books related to UML = 20.)

UML

Unified Modelling Language

1980's = worst of the “METHOD WARS”

In mid 1990's there were 3 main survivors

Grady Booch - “Booch Method”

James Rumbaugh - “Object Modeling
Technique” OMT

Ivar Jacobson - “Object-Oriented Software
Engineering” OOSE

These “THREE AMIGOS” joined forces at
Rational to develop UML.

In 1997 – Version 1.0 of UML was submitted to
the Object Management Group (OMG)

www.omg.org

UML (cont.)

A UML consortium sprang up with members like DEC, HP, Intellicorp, Microsoft, Oracal, TI, Rational and others. Currently at version 2.0 (See <http://www.omg.org>).

UML (cont.)

What UML is:

UML consists of a number of graphical elements that combine to form diagrams.

What UML is not:

UML does NOT specify a particular process.

Often, since it was developed at Rational, it is described along with the Rational Unified Process. Others have special names like GRAPPLE (Guidelines for Rapid APPLication Engineering) and FDD (Feature Driven Development)

UML (cont.)

Which process you use will depend on which company you chose to work for. For this UML description, we will use the unified process described in many books.

UML Components

The purpose of the UML diagrams is to present multiple views of a system; this set of multiple views is called a MODEL. The UML model of a system tells WHAT the system is supposed to do, NOT how to implement the system.

For our purposes, a model is a set of UML diagrams that we can examine, assess, and modify in order to understand and develop a system.

Class

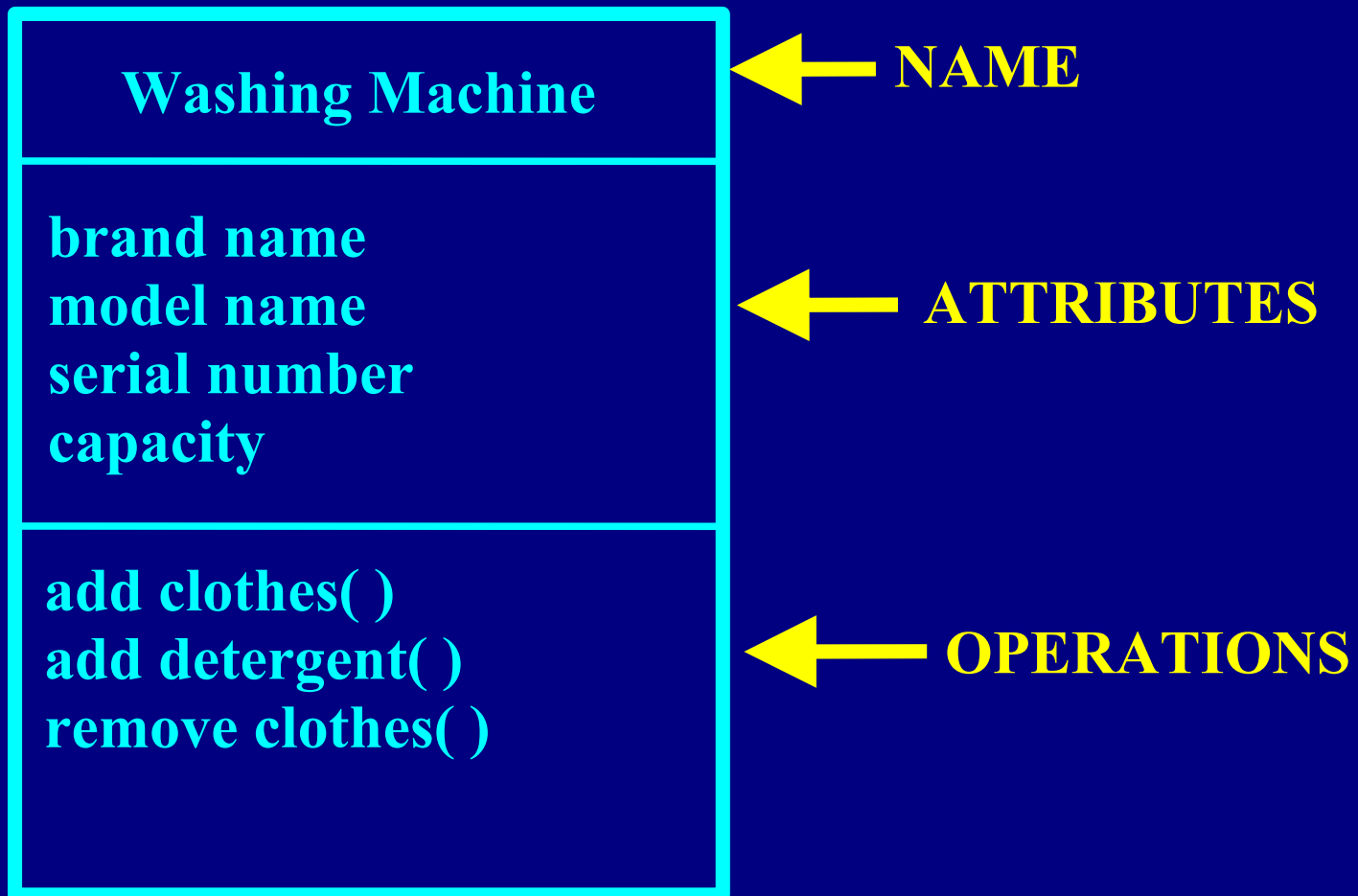
A class is a category or group of things that have similar attributes and common behaviors.

e.g. class of washing machines.

Attributes: brand name, model, serial number, and capacity.

Behaviors: “add clothes”, “add detergent”, “turn on”, and “remove clothes”.

UML Class Icon



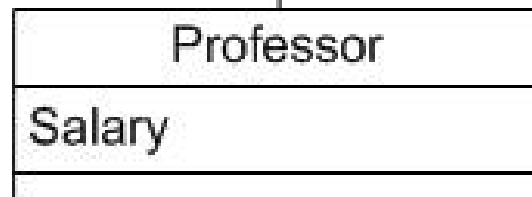
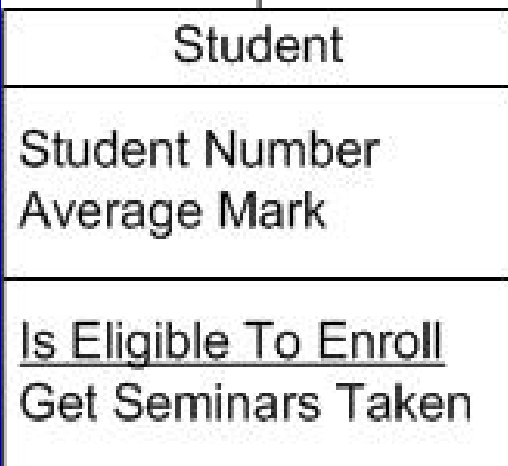
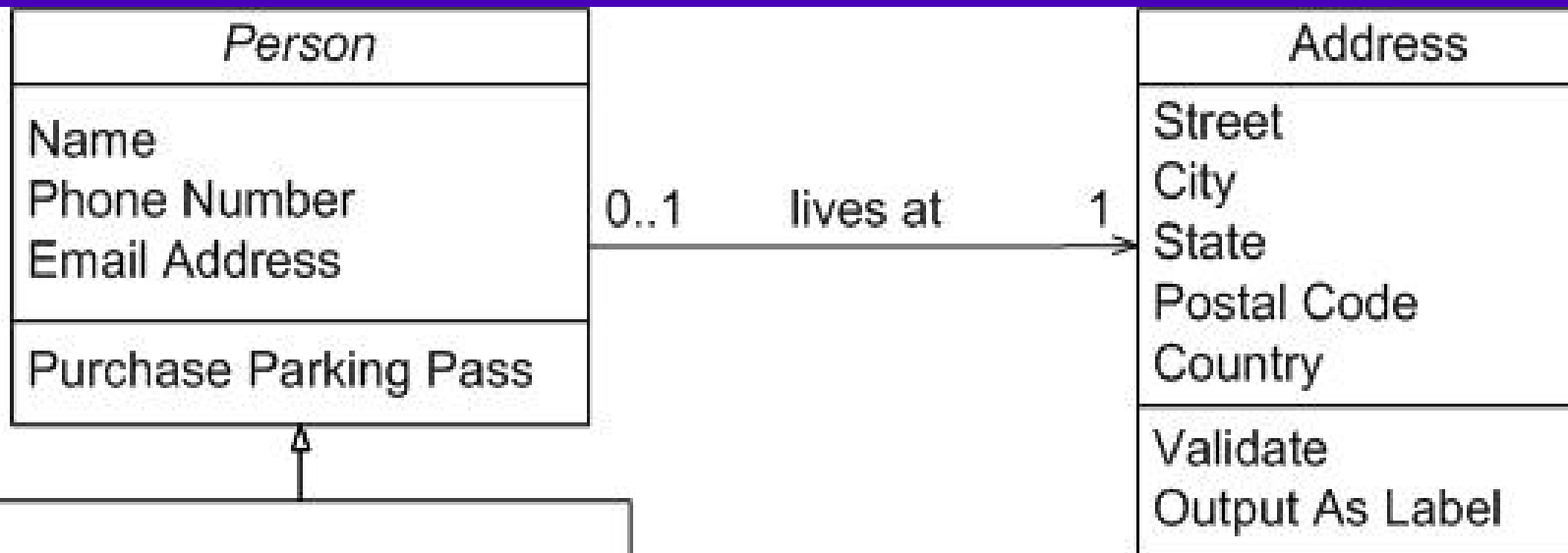
Class Diagram

A class diagram consists of a number of these class icons connected by lines that show how the classes relate to one another.

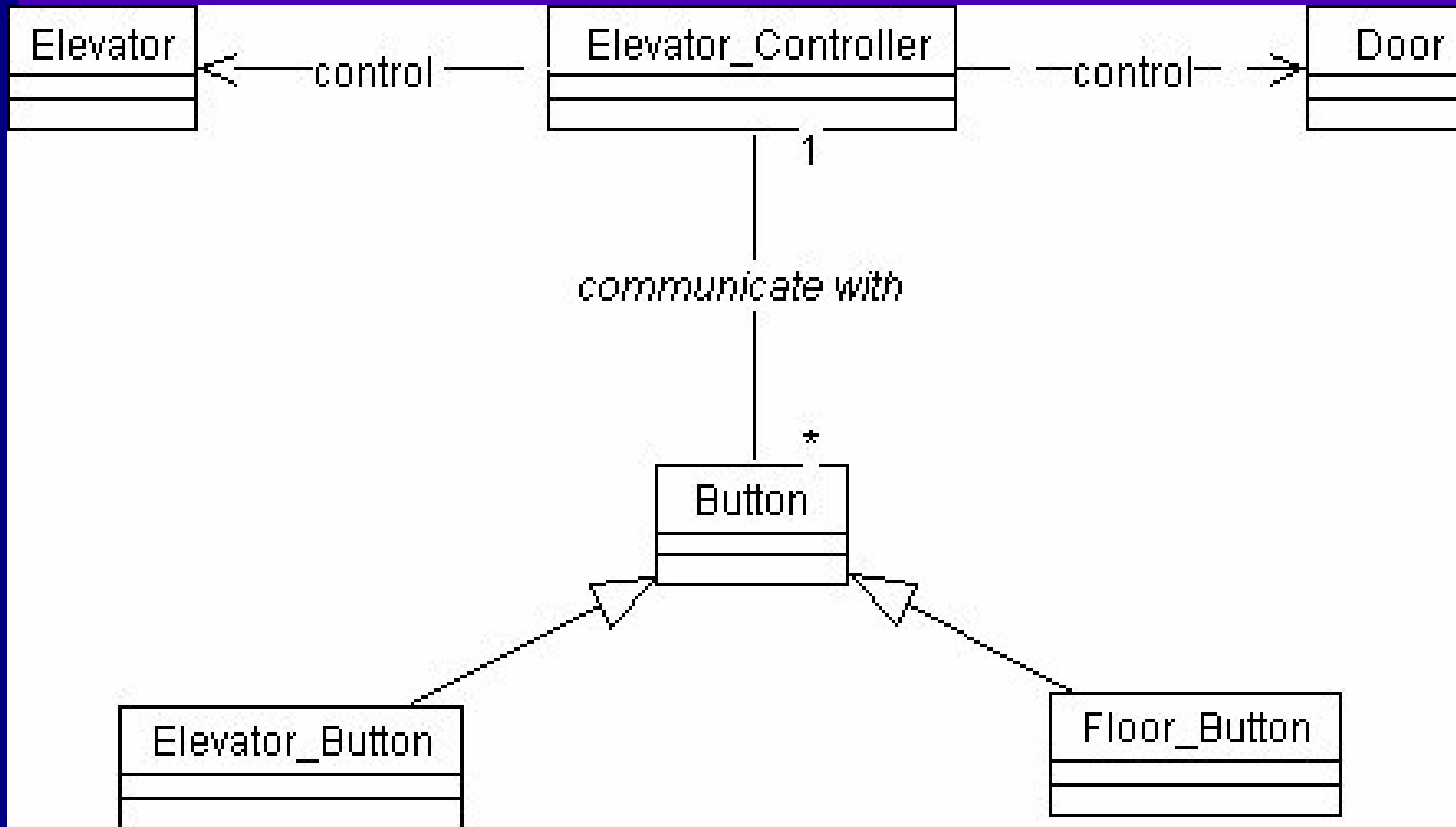
Page 99 in Stiller/LeBlanc book shows a class diagram.

Another example class diagram is on the next slide.

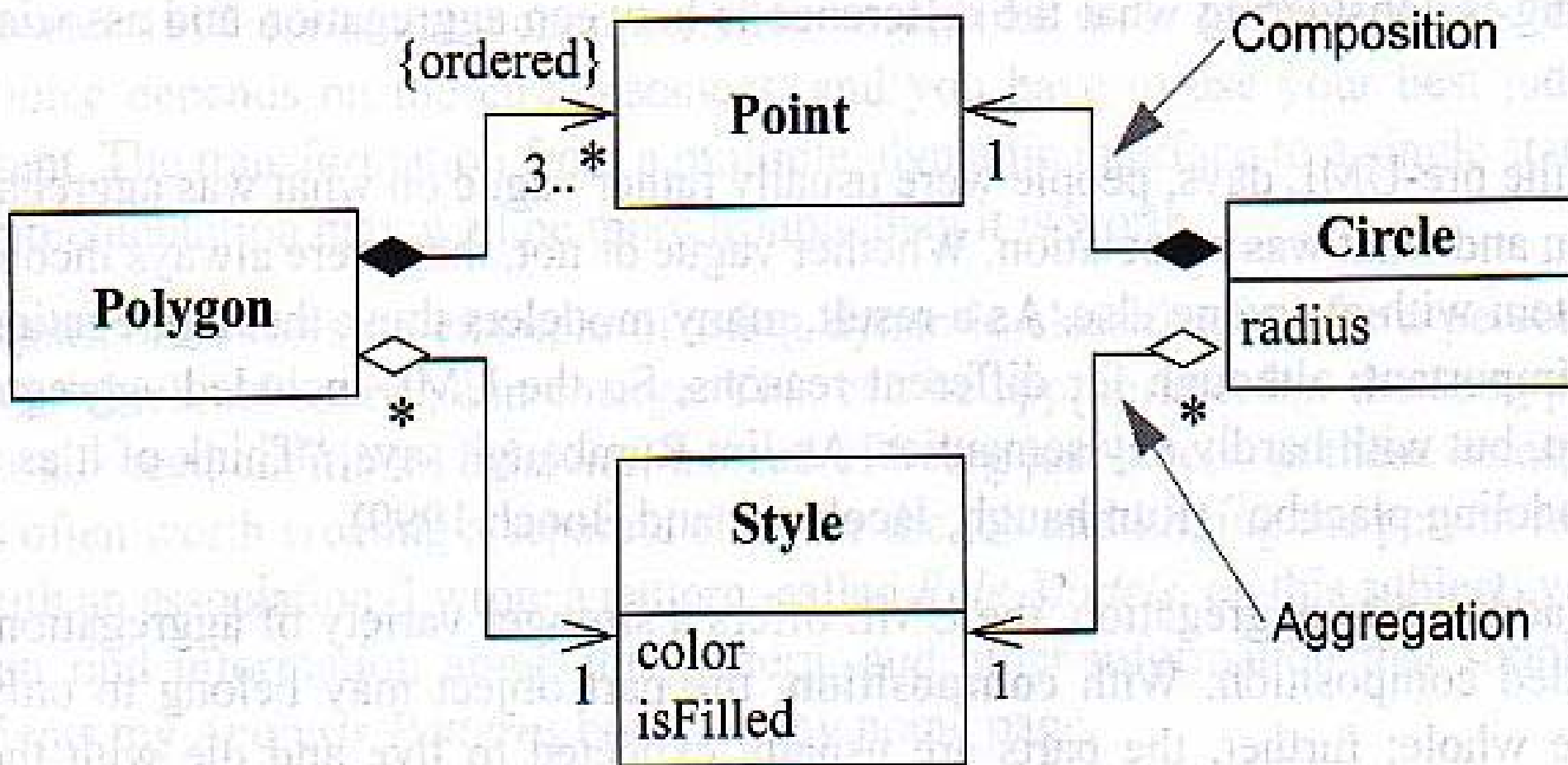
Example Class Diagram



Example Class Diagram



Example Class Diagram



Object Diagram

An object diagram is an instance of a class
– a specific thing that has specific values
of the attributes and behaviors.

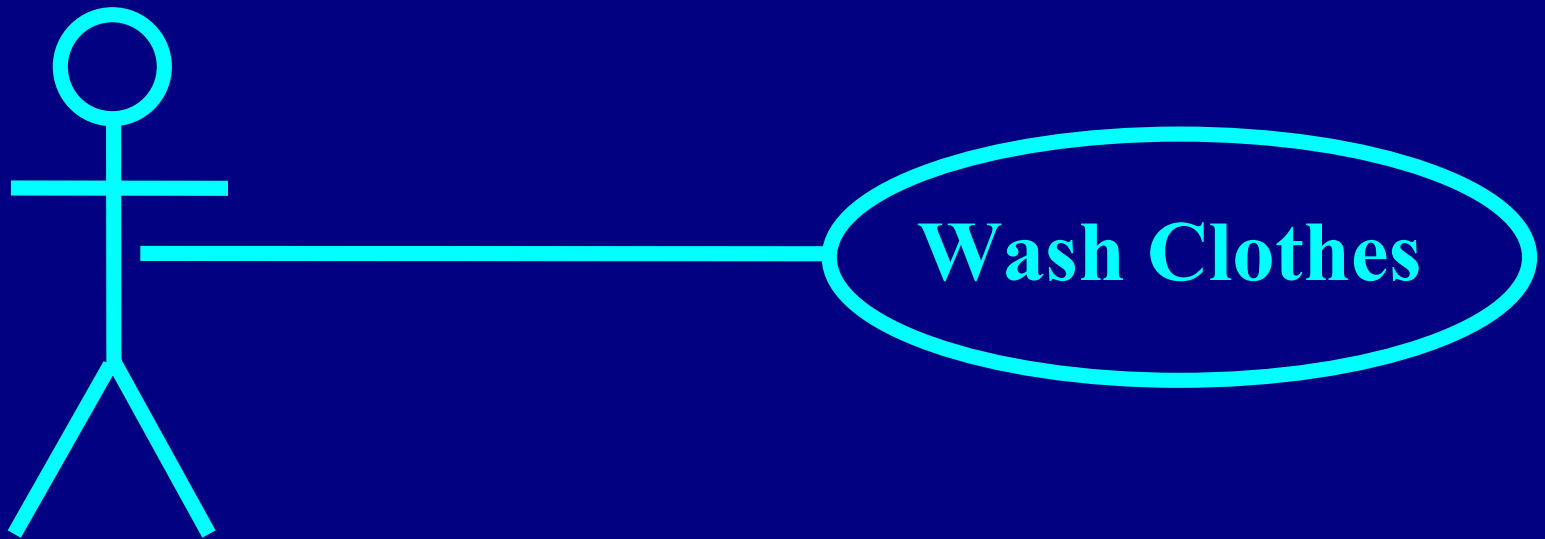
A UML Object Diagram icon, which is a rectangle with a thick black border. Inside the rectangle, the text "My Washer : Washing Machine" is underlined.

My Washer : Washing Machine

The icon is a rectangle like the class, but
the name is underlined, see page 117
In SL.

Use Case Diagram

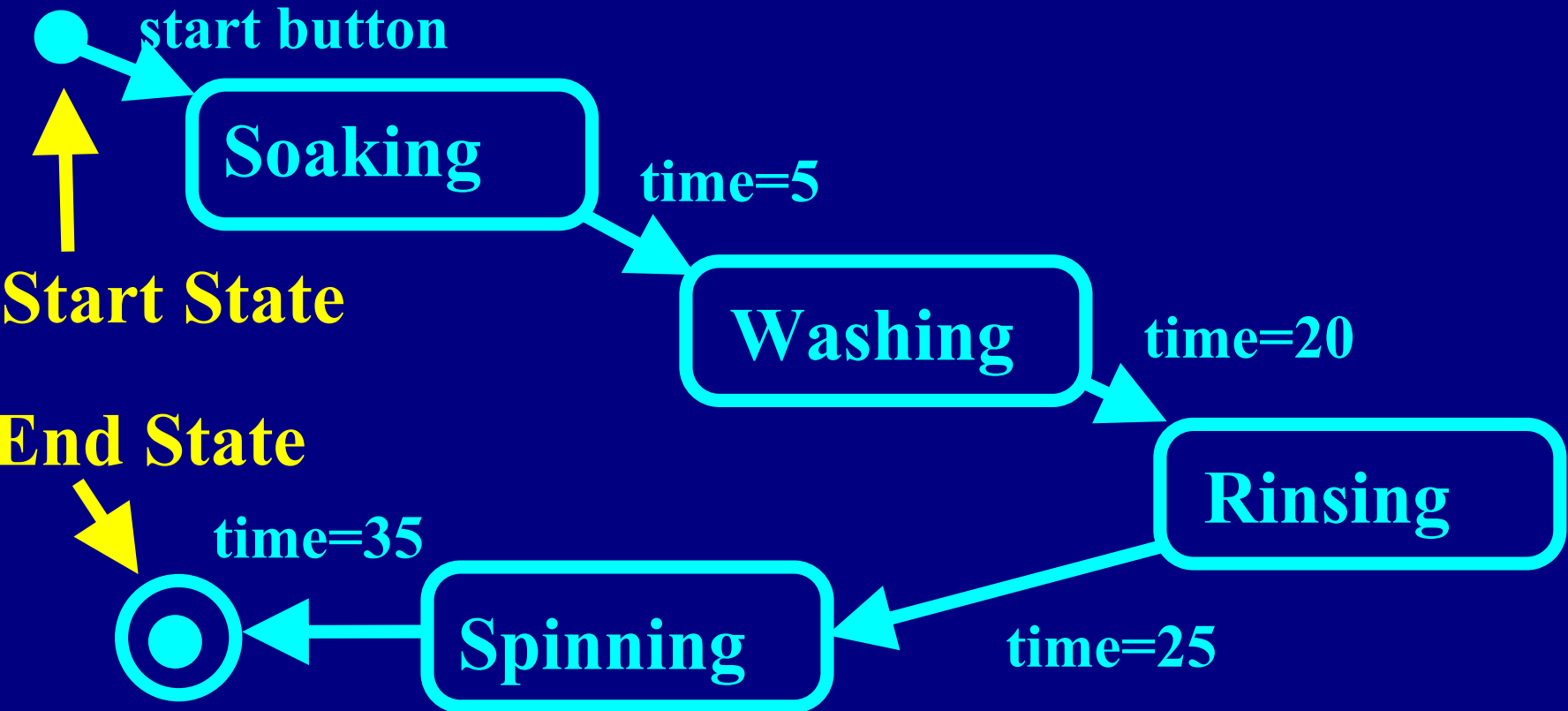
A use case is a description of a system's behavior from a user's standpoint.



Washing Machine User

State Diagram

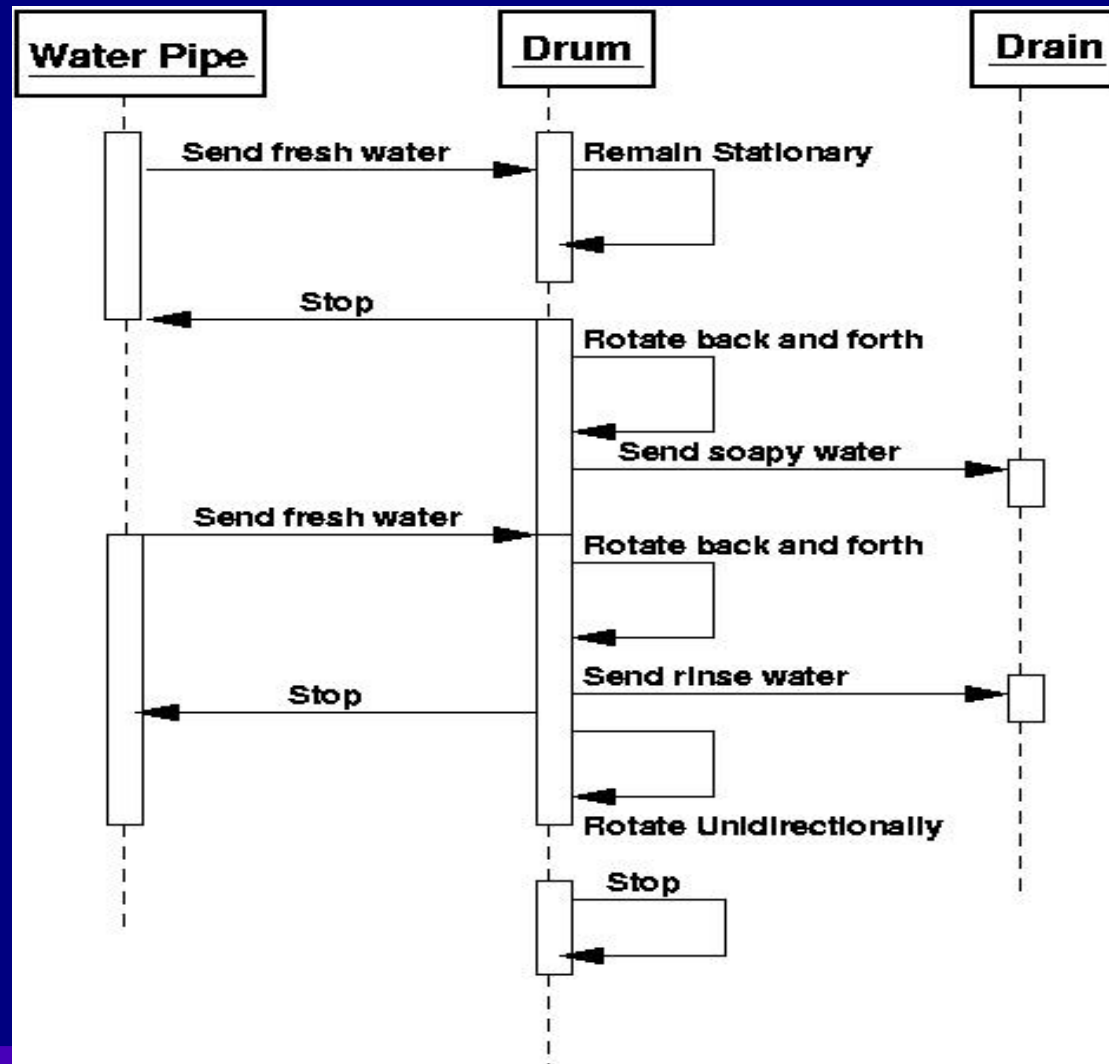
At any time, an object is in a particular state.



Sequence Diagram

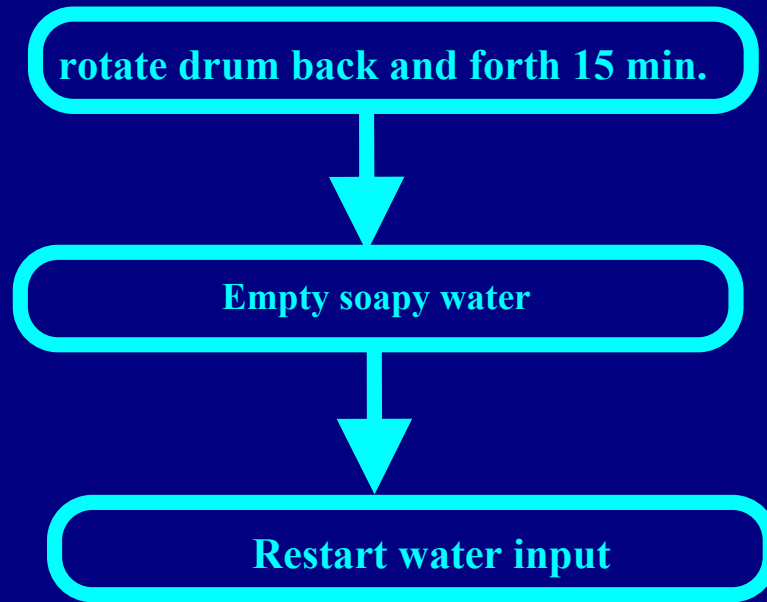
The sequence diagram shows the time-based dynamics of the interaction between objects.

Example Sequence Diagram



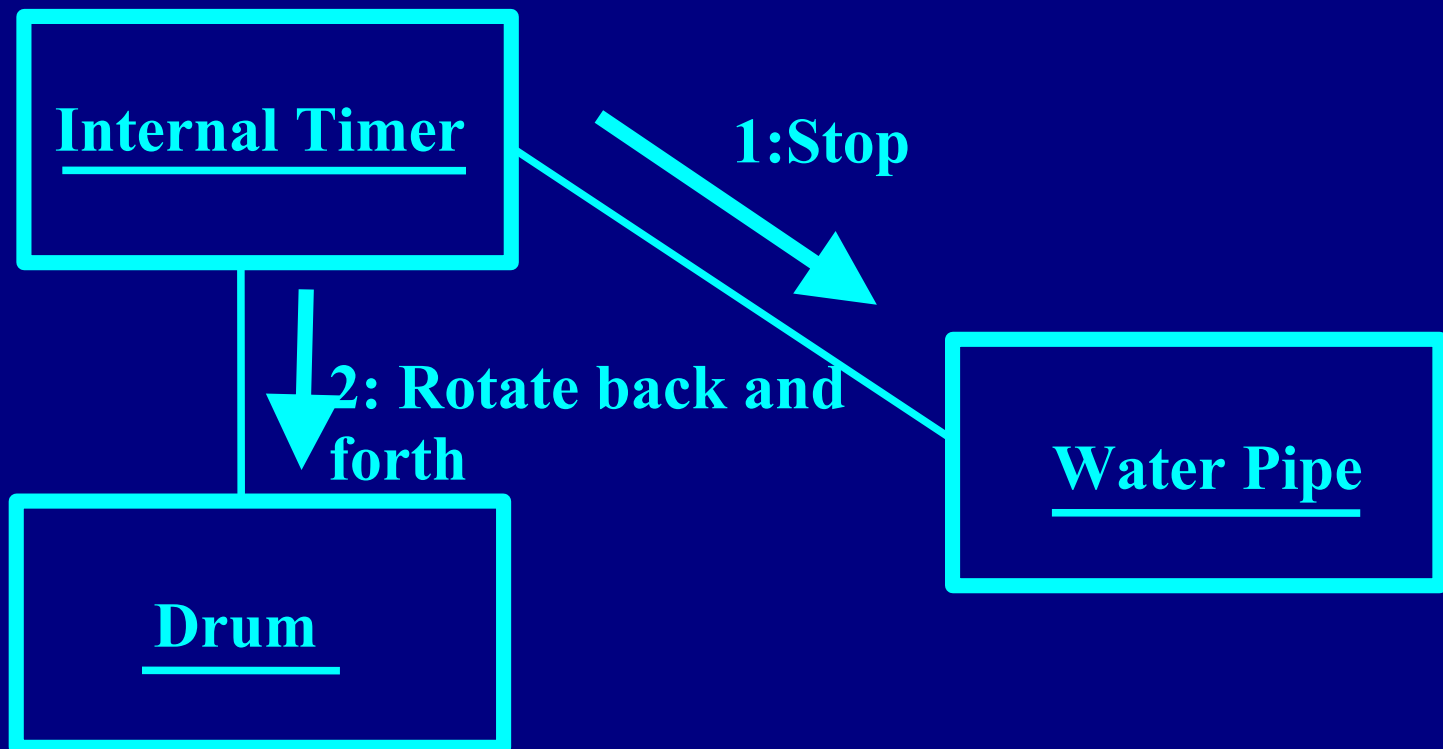
Activity Diagram

An activity diagram is used to represent steps in a sequence.



Collaboration Diagram

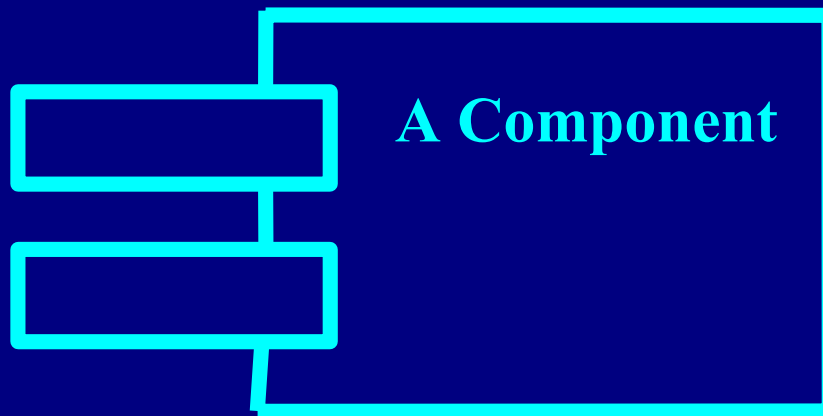
The collaboration diagram shows how the elements of a system work together.



Component Diagram

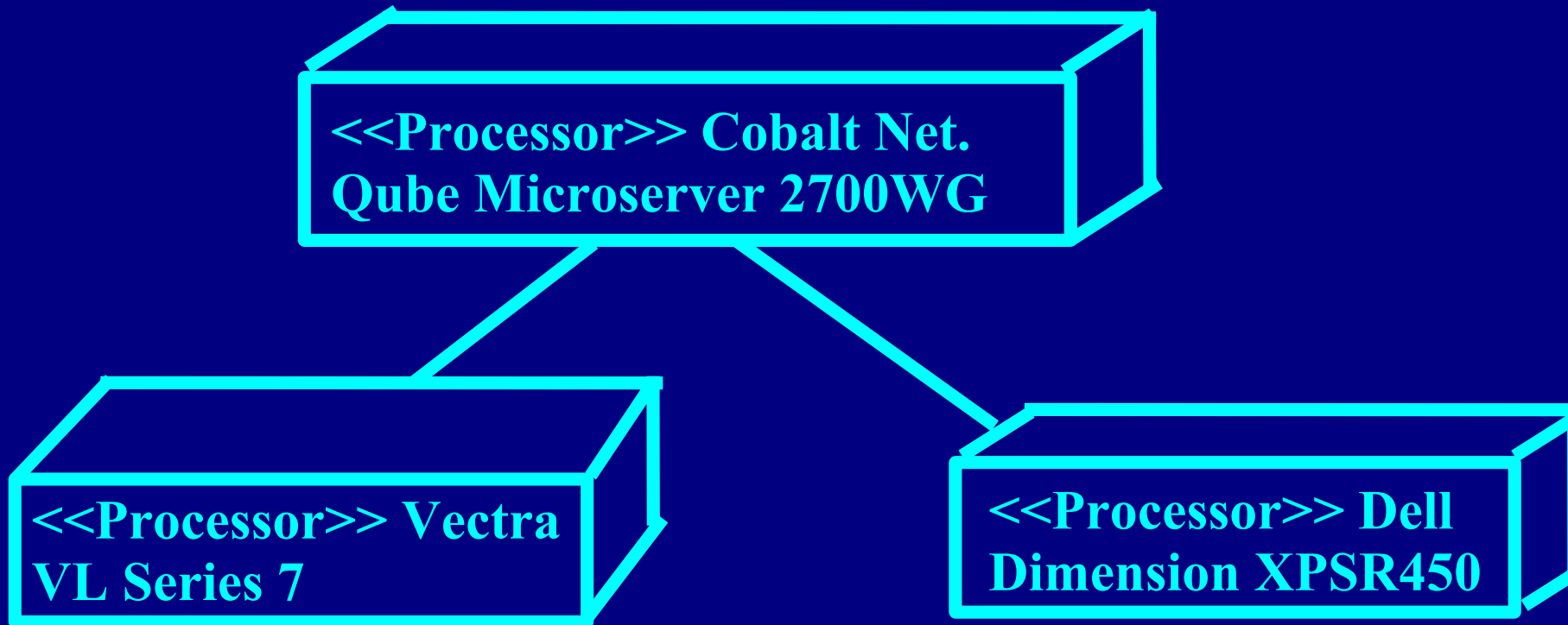
A component is a physical and replaceable part of a system that conforms to, and realizes, a set of interfaces.

(not a good analogy in the washing machine)



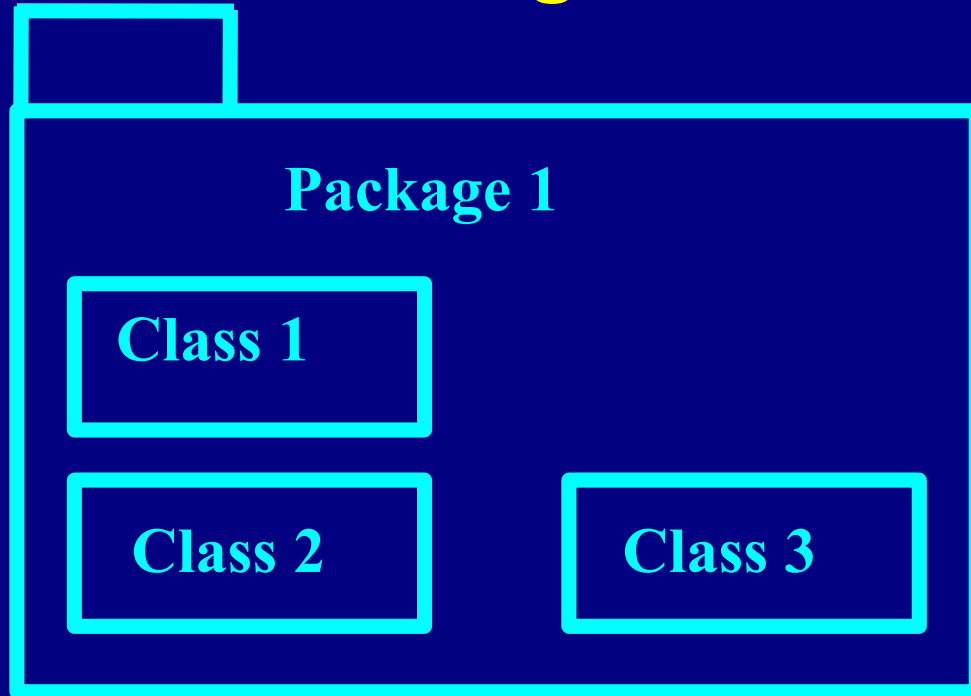
Deployment Diagram

The UML deployment diagram shows the physical architecture of a computer-based system.



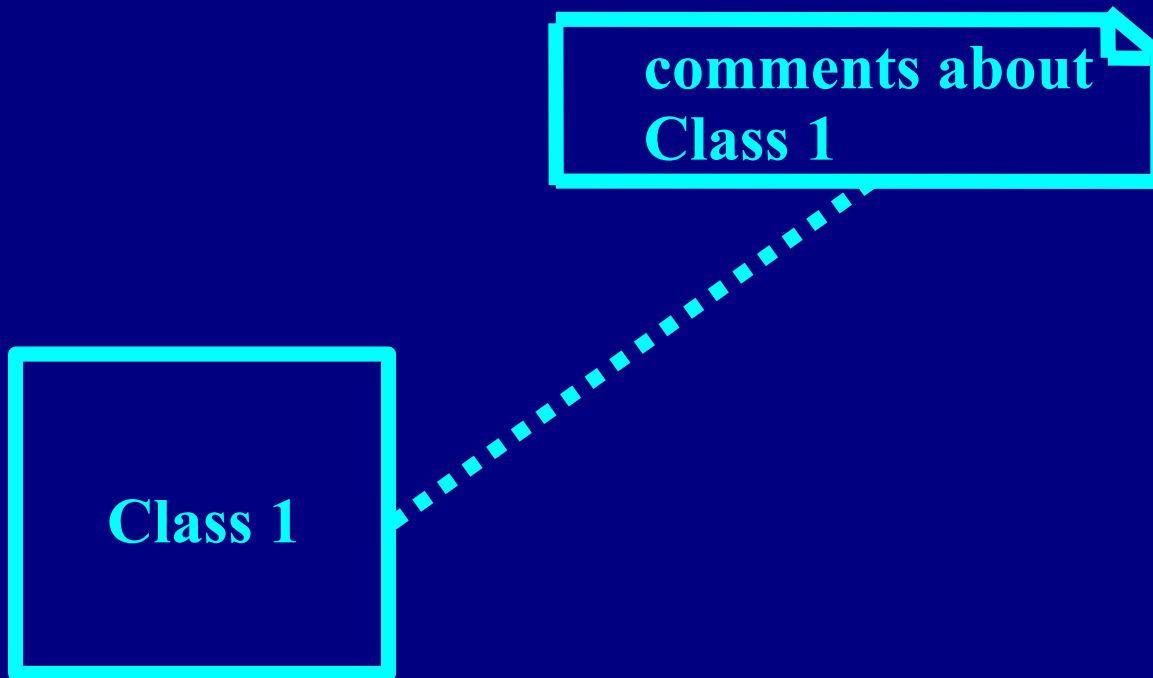
Packages

Sometimes you'll find the need to organize the elements of a diagram into a group.



Notes

“Comments” in a diagram are done using a notes diagram.



Object Oriented (revisited)

Abstraction

Inheritance

Polymorphism

Encapsulation

Message Sending

Associations

Aggregation

Abstraction

Abstraction means, simply, to filter out an object's properties and operations until just the ones you need are left.

The abstraction level can change depending on your function. Think of an automobile from the following:

Driver

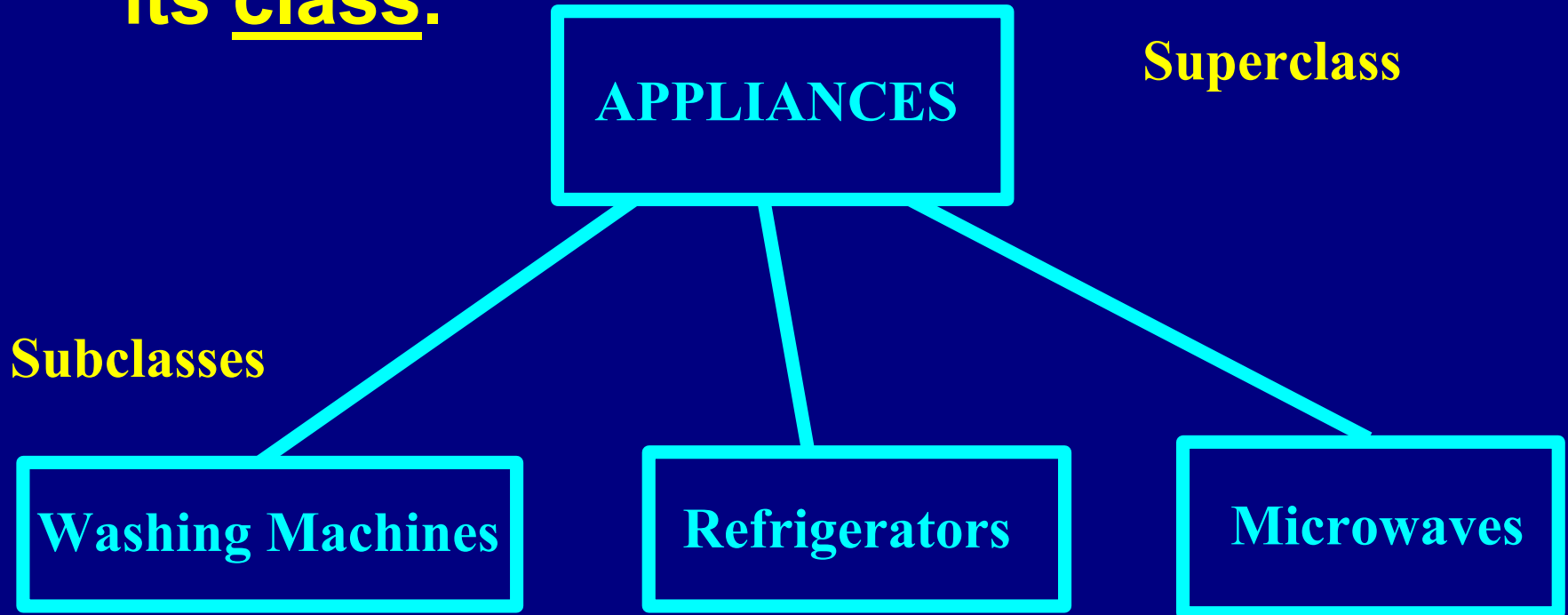
Designer

Mechanic

Police Officer

Inheritance

An object inherits all the characteristics of its class.



Polymorphism

Sometimes an operation has the same name in different classes. For example, OPEN.

Open a door

Open a bank account

Open a window

Open a package

Open a newspaper

Open a conversation

The class knows how to respond to the open message.

Encapsulation

In a TV commercial from a few years back, two people discuss all the money they'll save if they dial a particular 7-digit prefix before dialing a long-distance phone call. One of them asks, incredulously, “How does it work?”

The other replies: “How does popcorn pop? Who cares?”

An object carries out its operations hidden from the rest of the world.

Message Sending

In a system, objects work together. They do this by sending messages to one another. One object sends another a message to perform an operation, and the receiving object performs that operation.

Think of a TV remote control. You could get up and do it yourself at the TV (but you'd be drummed out of the couch potato society). The remote is another “interface” for the TV operations.

Association

Objects are typically related to one another in some fashion. When using the remote control, there is a one-way association between you and your TV.

Objects may have many other associations.

Tom is a co-worker of Suzie.

Tom is a friend of Suzie.

An object may be associated with more than one other object.

Fred rides in a bus.

Fred rides in a car.

Aggregation

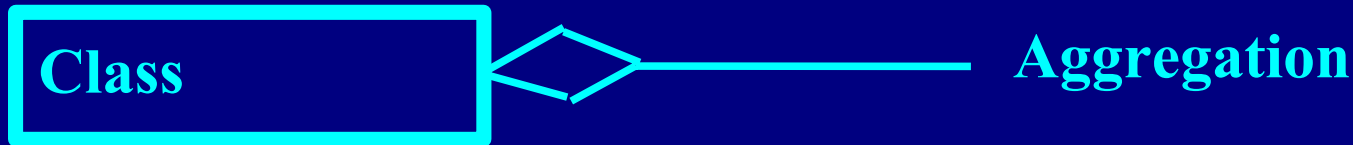
Your computer is an aggregation of a CPU box, a keyboard, a mouse, a monitor, a CD-ROM driver, one or more hard drives, a modem, a network card, a printer and possibly some speakers.

Your computer is “made up” from a number of different types of components.

If doing away with one of the parts invalidates the whole, you have a composition.

Aggregation/Composition Diagram Element

When you diagram an aggregation, you put an open diamond on the end of the line connecting the objects.



When you diagram a composition, you put a closed diamond on the end of the line connecting the objects.

