Chapter 3

Object Interaction

Solving Problems
Modularization

As problems become more complex, a single, simple solution becomes difficult, and then impossible.

To allow us to wrap our minds around a complex problem, we break it into smaller and smaller individual problems until we are dealing with a collection of simple problems.

This process is called modularization, and the process of creating modules is often referred to as a divide and conquer strategy.

Solving Problems
Abstraction

Now that the problem is broken into simpler pieces, we solve each piece individually.

As each smaller problem is solved, we move on to the next. Once a solution works, we ignore how it works, and incorporate that solution into our larger solution.

The ability to ignore the details of a solution, and focus on the bigger picture, is known as abstraction.

Solving Problems
Classes & Objects

To a programmer using an object-oriented approach (that's us!), solutions to simple problems are often created using a class.

It follows that a more complex problem may require multiple classes.

As we build simple classes to solve simple problems, they can be combined and interact with each other to solve a much larger problem.

The Clock Display

00:00 to 23:59

A digital clock has four digits. There are 60 minutes in an hour and 24 hours in a day.

The digits must progress in an orderly way, and each digit has a maximum value. Once it reaches the maximum, it returns to zero.

The Clock Display

00:00 to 23:59

Minutes:

The minutes can be thought of as a two-digit display, with a range of values from 00 to 59.

The Clock Display

00:00 to 23:59

Hours:

The hours can be thought of as a two-digit display, with a range of values from 00 to 23.

Implementing the Clock Display

```
public class NumberDisplay
{
    private int limit;
    private int value;

    // constructor & methods ommitted
}
```

This class will be responsible for counting up to the specified limit and then rolling over to zero.

Implementing the Clock Display

```
public class ClockDisplay
{
    private NumberDisplay hours;
    private NumberDisplay minutes;

    // constructor & methods ommitted
}
```

We have created fields that have classes as data types! A field stores data... these fields store objects.

Class Diagrams

Class diagrams show the classes in a system and their relationship to each other. Only one representation of each class is shown in the diagram.

Object diagrams show the relationship between objects. Since multiple objects can be created from the same class, the object diagram must show each object.

The object diagrams show the *runtime* view of the system.

Assigned Work

Read Chapter 3:
3.1 to 3.6 (Class diagrams vs object diagrams)

Complete exercises 3.1 to 3.4

Record your answers in a text document or OpenOffice document