Chapter 3
Object Interaction

3.8.4 to 3.10

Creating Objects
Example of Abstraction
Method Overloading

3.8: updateDisplay() method An example of abstraction

Abstraction – the ability to ignore the details of a solution and focus on the bigger picture

In the ClockDisplay class, there is a method updateDisplay. In the current program, the method formats and stores a string with the current time.

If we had actual clock hardware (e.g., LCD display), the method would still be used, but have very different code to perform its job.

3.9: Storing by Reference

When an object data type is declared, the compiler requests space in RAM to store the *address* of the variable, which will actually exist somewhere else in RAM.

```
    private NumberDisplay hours;
    private NumberDisplay minutes;
```

So far, we have created a reference (or address, or pointer) to a NumberDisplay object, but the object doesn't actually exist yet.

3.9: Creating an Object in Memory

```
public ClockDisplay()
{
    hours = new NumberDisplay(24);
    minutes = new NumberDisplay(60);
    updateDisplay();
}
```

The new operation creates an object of the specified class in memory and executes the constructor for that class (with appropriate parameters).

3.9: Creating an Object in Memory

In general,

(a) declare the object variable

```
        private ClassName variableName;
```

(b) create the object and assign to a variable

```
        variableName = new ClassName ( ... );
```

3.10: Method Overloading

The ClockDisplay class has two constructors: one with zero parameters, one with two parameters.

```
public ClockDisplay()
public ClockDisplay(int hour, int minute)
```

It is possible to have multiple methods with the same name, provided they have different parameter signatures (i.e., the number and *type* of parameters must be unique for each method).

3.10: Method Overloading

For example, the following would be permitted:

```
public Volume()
public Volume(int x)
public Volume(float x)
public Volume(int x, int y, int z)
```

This would produce an error:

```
public Volume(int x, float y)
public Volume(int alpha, float beta)
```

Assigned Work

Read Chapter 3: 3.8.4 to 3.10

Complete exercises to 3.27
(and catch up on previous work)

Record your answers in a text document or OpenOffice document