CECS 277 – Lecture 1 – Classes Review

Classes – A class is a blueprint for an object. It specifies the fields and methods a particular type of object has. One or more objects are then made from the class.

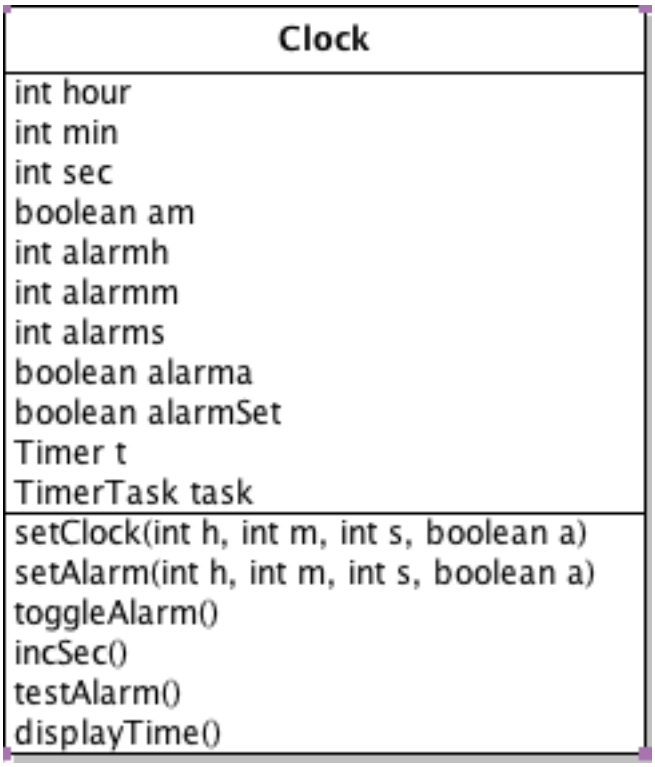
Classes are particularly useful to help maintain larger projects, they help cut down on excess code by being reusable, and they keep code encapsulated. This allows the programmer to be ensured that once the class has been written and tested, that the class doesn't need to be maintained or constantly changed. The class can be used to make objects or other classes by being extended to the new class.

A class is made up of data members and methods. Data members store information about the object, and methods are created to allow the object to be functional.

Example – Alarm Clock Class – An alarm clock has several features and functions.

Features: Stores the time in hours, minutes, seconds, and whether it is am or pm. It also stores an alarm time in hours, minutes, seconds, and am or pm, along with whether the alarm is set to on or off.

Functions: An alarm clock allows you to set the time, set the alarm time, turn the alarm on or off, and to see the displayed time. Other functions, which are not implemented by the user (private functions) might include, incrementing the time, and sounding the alarm.



```java
import java.util.*;
/**
 *  Clock is a simple representation of an Alarm Clock
 *  @author Shannon Foss
 *  @version Date: Jan 24, 2012
 */
public class Clock {
     /** Clock hour - 0-11 */
     private int hour;
     /** Clock minute - 0-59 */
     private int min;
     /** Clock second - 0-59 */
     private int sec;
     /** Part of day - am=true, pm=false */
     private boolean am;

     /** Alarm hour - 0-11 */
     private int almHour;
     /** Alarm minute - 0-59 */
     private int almMin;
     /** Alarm second - 0-59 */
     private int almSec;
     /** Part of day for alarm - am=true, pm=false */
     private boolean almAm;//true is am

     /** Alarm Set? Yes=true, no=false */
     private boolean alarmSet;//true is set

     /** Timer object - to count seconds */
     private Timer t;
     /** TimerTask object - to call the incSec() method */
     private TimerTask task;

     /** Initializes clock and alarm to 00:00:00am.
      *    Initializes alarm to off.
      *    Creates a new timer and task.
      *    Schedules the task to be performed once per second.
      */
     public Clock(){
          hour = 0;
          min = 0;
          sec = 0;
          am = true;
          almHour = 0;
```

```
		almMin = 0;
		almSec = 0;
		almAm = true;
		alarmSet = false;
		t = new Timer();
		task = new TimerTask(){
			public void run(){
				incSec();
				displayTime();
				testAlarm();
			}
		};
		t.schedule(task,0, 1000);
	}
	/** Allows the clock's time to be set.
	 *	@param h  The hour the clock is going to be set to.
	 *	@param m  The minute the clock is going to be set to.
	 *	@param s  The second the clock is going to be set to.
	 *	@param a  The part of the day the clock is going to
	be set to.
	 */
	public void setClock(int h, int m, int s, boolean a){
		hour = h;
		min = m;
		sec = s;
		am = a;
	}
	/** Allows the alarm's time to be set.
	 *	@param h  The hour the alarm is going to be set to.
	 *	@param m  The minute the alarm is going to be set to.
	 *	@param s  The second the alarm is going to be set to.
	 *	@param a  The part of the day the alarm is going to
	be set to.
	 */
	public void setAlarm(int h, int m, int s, boolean a){
		almHour = h;
		almMin = m;
		almSec = s;
		almAm = a;
	}
	/** Turns the alarm on or off. */
	public void toggleAlarmSwitch(){
		alarmSet = !alarmSet;
	}
```

```
/** Increments the seconds. */
private void incSec(){
      sec++;
      if (sec == 60){
            sec = 0;
            min++;
            if (min == 60){
                  min = 0;
                  hour++;
                  if(hour == 12){
                        am = !am;
                        hour=0;
                  }
            }
      }
}
/** Tests to see if the alarm time has been reached. */
private void testAlarm(){
      if(alarmSet && hour == almHour && min == almMin && sec
      == almSec && am == almAm){
            System.out.println("BEEP BEEP BEEP");
      }
}
/** Displays the formatted time. */
public void displayTime(){
      System.out.print("The Time is: ");
      if(hour == 0){
            System.out.print("12:");
      }else{
            System.out.print(hour+":");
      }
      if(min < 10){
            System.out.print("0");
      }
      System.out.print(min+":");
      if(sec < 10){
            System.out.print("0");
      }
      System.out.print(sec+" ");
      if(am){
            System.out.println("am");
      }else{
            System.out.println("pm");
      }
```

```
	}
}
/*	Author: Shannon Foss
	Date: Jan 24, 2012
	Program: TestAlarm.java - Tester for Clock class.
	Desc: Example of writing a class.
*/
public class TestAlarm {
	public static void main(String[] args) {
		Clock c = new Clock();
		c.setClock(12, 0, 0, true);
		c.setAlarm(12, 1, 0, true);
		c.toggleAlarmSwitch();
	}
}
/*Output
The Time is: 12:00:01 am
The Time is: 12:00:02 am
The Time is: 12:00:03 am
The Time is: 12:00:04 am
...
The Time is: 12:00:59 am
The Time is: 12:01:00 am
BEEP BEEP BEEP
The Time is: 12:01:01 am
...
*/
```