```
/**
 * This class is intended to model a 12-hour digital clock (e.g. 07:34:52 or
12:02:59).
 * The clock is able to be set to a specific time and then accurately begin keeping
time.
 * When appropriate, the time rolls over to the next minute, next hour, or both.
 *
 * @author Alexander Fanning
 */
public class DigitalClock
{
    // Fields/instance variables
    private int hour;
    private int minute;
    private int second;
    private boolean hourMode;

    /**
     * Constructor for objects of class DigitalClock
     * Replace the constructor.
     * Rather than assigning the fields with the parameters in 3 different
statements,
     * make a call to the setTime method using the constructor's parameters as the
     * setTime method's parameters
     */
    public DigitalClock(int h, int m, int s)
    {
        setTime(h, m, s);
    }

    /**
     * Constructor for objects of class DigitalClock
     * Replace the constructor. Set the corresponding field to the value of mode.
     * Rather than assigning the fields with the parameters in 3 different
statements,
     * make a call to the setTime method using the constructor's parameters as the
     * setTime method's parameters
     */
    public DigitalClock(int h, int m, int s, boolean mode)
    {
        hourMode = mode;
        setTime(h, m, s);
    }

    /**
     * Assigns the fields by calling the appropriate set methods.
     * In order to set the time, you must set the hour, set the minute, and set the
second.
     */
    public void setTime(int h, int m, int s)
    {
        setHour(h);
        setMinute(m);
        setSecond(s);

    }

    /**
     * Mutator method for the hour field.
```

```
 * It should check if the parameter is valid. If the parameter is invalid,
 * assign hour to a value of 1.
 * Fill in below.
 */
public void setHour(int h)
{
    if(hourMode == true && (h < 0 || h >= 25)){
        h = hour = 0;
    }
    else if(hourMode == false && (h <= 0 || h >= 13)){
        h = hour = 1;
    }
    else{
        hour = h;
    }
}

/**
 * Mutator method for the minute field.
 * It should check if the parameter is valid. If the parameter is invalid,
 * assign minute to a value of 0.
 * Fill in below.
 */
public void setMinute(int m)
{
    if(m < 0 || m >= 60){
        m = minute = 0;
    }
    else{
        minute = m;
    }
}

/**
 * Mutator method for the hour field.
 * It should check if the parameter is valid. If the parameter is invalid,
 * assign second to a value of 0.
 * Fill in below.
 */
public void setSecond(int s)
{
    if(s < 0 || s >= 60){
        s = second = 0;
    }
    else{
        second = s;
    }
}

/**
 * Update the hour field to the next hour.
 * Take note that nextHour() of 12:37:29 is 01:37:29 assuming 12-hour mode
 * and the nextHour() of 23:47:12 is 00:47:12 assuming 24-hour mode.
 */
public void nextHour()
{
    if(hourMode == true){
        hour = (hour + 1) % 24;
        setHour(hour);
```

```
    }
    else{
        hour = (hour + 1) % 13;
        setHour(hour);
    }
}

/**
 * Update the minute field to the next minute.
 * Take note that nextMinute() of 03:59:13 is 04:00:13.
 */
public void nextMinute()
{
    if(minute == 59){
        nextHour();
    }
    minute = (minute + 1) % 60;
    setMinute(minute);
}

/**
 * Update the second field to the next second.
 * Take note that nextSecond() of 23:59:59 is 00:00:00.
 */
public void nextSecond()
{
    if(second == 59){
        nextMinute();
    }

    second = (second + 1) % 60;
    setSecond(second);

}

/**
 * Accessor method for the hour field.
 * Replace below.
 */
public int getHour()
{
    return hour;
}

/**
 * Accessor method for the minute field.
 * Replace below.
 */
public int getMinute()
{ return minute; }

/**
 * Accessor method for the second field.
 * Replace below.
 */
public int getSecond()
{ return second; }

/**
```

```
     * returns "HH:MM:SS"
     * Hint: You might find it helpful to create local String variables.
     * Replace below.
     */
    @Override
    public String toString()
    {

        return String.format("%02d:%02d:%02d",hour,minute,second);
    }

}
```