

Solution to CSE143 Section #17 Problems

1. One possible solution appears below.

```
public class Date implements Comparable<Date> {
    private int month;
    private int day;
    private int year;
```

```
    public Date(int month, int day, int year) {
        this.month = month;
        this.day = day;
        this.year = year;
    }
```

```
    public int getMonth() {
        return month;
    }
```

```
    public int getDay() {
        return day;
    }
```

```
    public int getYear() {
        return year;
    }
```

```
    public String toString() {
        String result;
        if (month < 10) {
            result = "0" + month + "/";
        } else {
            result = month + "/";
        }
        if (day < 10) {
            result += "0" + day;
        } else {
            result += day;
        }
        result += "/" + year;
        return result;
    }
```

```
    public int compareTo(Date other) {
        if (year != other.year) {
            return (year - other.year);
        } else if (month != other.month) {
            return (month - other.month);
        } else {
            return (day - other.day);
        }
    }
```

```
}  
}
```

2. Two possible solutions appear below.

```
public class USCurrency implements Comparable<USCurrency> {  
    private int dollars;  
    private int cents;  
  
    public USCurrency(int dollars, int cents) {  
        int totalCents = 100 * dollars + cents;  
        this.dollars = totalCents / 100;  
        this.cents = totalCents % 100;  
    }  
  
    public int dollars() {  
        return dollars;  
    }  
  
    public int cents() {  
        return cents;  
    }  
  
    public String toString() {  
        String s = Math.abs(dollars) + ".";  
        if (Math.abs(cents) < 10) {  
            s += "0" + Math.abs(cents);  
        } else {  
            s += Math.abs(cents);  
        }  
        if (dollars < 0 || cents < 0) {  
            return "-$" + s;  
        } else {  
            return "$" + s;  
        }  
    }  
  
    public USCurrency add(USCurrency other) {  
        return new USCurrency(dollars + other.dollars,  
                               cents + other.cents);  
    }  
  
    public USCurrency subtract(USCurrency other) {  
        return new USCurrency(dollars - other.dollars,  
                               cents - other.cents);  
    }  
  
    public int compareTo(USCurrency other) {  
        return dollars * 100 + cents - other.dollars * 100 - other.cents;  
    }  
}  
public class USCurrency implements Comparable<USCurrency> {
```

```

private int totalCents;

public USCurrency(int dollars, int cents) {
    totalCents = dollars * 100 + cents;
}

public int dollars() {
    return totalCents / 100;
}

public int cents() {
    return totalCents % 100;
}

public String toString() {
    int cents = Math.abs(totalCents);
    String s = cents / 100 + "." + cents % 100 / 10 + cents % 10;
    if (totalCents < 0) {
        return "-$" + s;
    } else {
        return "$" + s;
    }
}

public USCurrency add(USCurrency other) {
    return new USCurrency(dollars() + other.dollars(),
        cents() + other.cents());
}

public USCurrency subtract(USCurrency other) {
    return new USCurrency(dollars() - other.dollars(),
        cents() - other.cents());
}

public int compareTo(USCurrency other) {
    return totalCents - other.totalCents;
}
}

```

3. One possible solution appears below.

```

public class ClockTime implements Comparable<ClockTime> {
    private int hours;
    private int minutes;
    private String amPm;

    public ClockTime(int hours, int minutes, String amPm) {
        this.hours = hours;
        this.minutes = minutes;
        this.amPm = amPm;
    }
}

```

```

public int compareTo(ClockTime other) {
    if (!amPm.equals(other.amPm)) {
        return amPm.compareTo(other.amPm);
    } else if (hours != other.hours) {
        return hours % 12 - other.hours % 12;
    } else {
        return minutes - other.minutes;
    }
}

public int getHours() {
    return hours;
}

public int getMinutes() {
    return minutes;
}

public String getAmPm() {
    return amPm;
}

public String toString() {
    String result = hours + ":";
    if (minutes < 10) {
        result += "0" + minutes;
    } else {
        result += minutes;
    }
    result += " " + amPm;
    return result;
}
}

```

4. One possible solution appears below.

```

public class BookData implements Comparable<BookData> {
    private String title;
    private String author;
    private int reviews;
    private double total;

    public BookData(String title, String author) {
        this.title = title;
        this.author = author;
        this.reviews = 0;
        this.total = 0.0;
    }

    public void review(double rating) {

```

```

        reviews++;
        total += rating;
    }

    public String getTitle() {
        return title;
    }

    public double getRating() {
        if (reviews == 0) {
            return 0.0;
        } else {
            return total / reviews;
        }
    }

    public String toString() {
        double rating = (int) (10.0 * getRating()) / 10.0;
        String result = title + ", by " + author + ", " + rating + " (";
        if (reviews == 1) {
            result += "1 review)";
        } else {
            result += reviews + " reviews)";
        }
        return result;
    }

    public int compareTo(BookData other) {
        double delta = getRating() - other.getRating();
        if (delta < 0) {
            return 1;
        } else if (delta > 0) {
            return -1;
        } else // delta == 0 {
            return other.reviews - reviews;
        }
    }
}

```