CSE 142 Section Handout #4
Problems

if/else

(Ch. 4 self-checks 1-12)

1. ("If/Else Mystery"). For each call to the following method, indicate what output is produced.

```
public static void mystery1(int n) {
    System.out.print(n + " ");
    if (n > 0) {
        n = n - 5;
    }
    if (n < 0) {
        n = n + 7;
    } else {
        n = n * 2;
    }
    System.out.println(n);
}
```

Call	Output
`mystery1(8);`	_____
`mystery1(-3);`	_____
`mystery1(1);`	_____
`mystery1(0);`	_____

2. ("If/Else Mystery"). For each call to the following method, indicate what output is produced.

```
public static void mystery2(int a, int b) {
      if (a % b == 0) {
         a = a / b;
         if (a < b) {
            b = b - a;
         }
      } else if (b % 2 == 0) {
         b = b / 2;
      } else {
         a = a - b;
      }
      System.out.println(a + " " + b);
}
```

Call	Output
`mystery2(20, 4);`	_____
`mystery2(7, 6);`	_____
`mystery2(14, 7);`	_____
`mystery2(24, 8);`	_____
`mystery2(13, 9);`	_____

3. ("Coordinates"). Rewrite the code shown by **factoring** to eliminate redundancy. In other words, move common/repeated code so that it does not need to be written in multiple places.

```
Scanner console = new Scanner(System.in);
System.out.print("x coordinate? ");
double x = console.nextDouble();
int neg;
if (x < 0.0) {
    System.out.print("y coordinate? ");
    double y = console.nextDouble();
    if (y < 0.0) {
        neg = 2;
        System.out.println("negatives = " + neg);
    }  else {
        neg = 1;
        System.out.println("negatives = " + neg);
    }
} else {
    System.out.print("y coordinate? ");
    double y = console.nextDouble();
    if (y < 0.0) {
        neg = 1;
        System.out.println("negatives = " + neg);
    }  else {
        neg = 0;
        System.out.println("negatives = " + neg);
    }
}
```

(continued on back page)

CSE 142 Section Handout #4
Problems (continued)

`return` and `if/else` *(Ch. 3 self-checks 12-17, ex. 6-17; Ch. 4 self-checks 4-6, ex. 10-15; Ch. 5 ex. 11-13)*

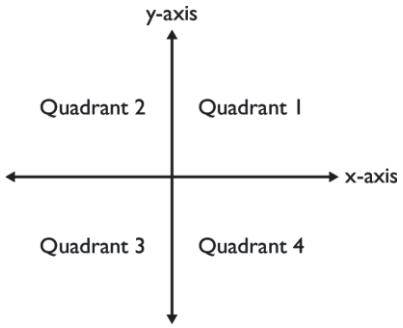
4. a) Exercise 4.4, p314. ("daysInMonth"). *(Tip: Try testing your solution in our Practice-It web system.)* Write a method named `daysInMonth` that accepts a month (an integer between 1 and 12) as a parameter and returns the number of days in that month. For example, the call `daysInMonth(9)` returns `30` because September has 30 days. Ignore leap years; assume that February always has 28 days.

Month	1 Jan	2 Feb	3 Mar	4 Apr	5 May	6 Jun	7 Jul	8 Aug	9 Sep	10 Oct	11 Nov	12 Dec
Days	31	28	31	30	31	30	31	31	30	31	30	31

b) Write a main method that prompts the user for a month (entered as a number) and passes that value into `daysInMonth`. It should print out the number of days returned from `daysInMonth`.

```
Enter a month (as an int): 9
There are 30 days in that month!
```

5. a) Exercise 4.19, p318 ("quadrant"). *(Tip: Try testing your solution in our Practice-It web system.)* Write a method called `quadrant` that accepts as parameters a pair of real numbers representing an (*x*, *y*) point and returns the quadrant number for that point. Quadrants are numbered as integers from 1 to 4 with the upper-right quadrant numbered 1 and the subsequent quadrants numbered in a counterclockwise fashion:



Notice that the quadrant is determined by whether the *x* and *y* coordinates are positive or negative numbers. Return `0` if the point lies on the *x*-axis or *y*-axis. For example, the call of `quadrant(-2.3, 3.5)` should return `2` and the call of `quadrant(7.1, -4.6)` should return `4` .

b) Write a main method that prompts the user for an x coordinate and a y coordinate and passes those values into `quadrant`. It should print out the quadrant number returned from `quadrant`.

```
X coordinate? 1.0
Y coordinate? 2.0
(1.0, 2.0) is in quadrant 1
```

`Scanner` and cumulative sum *(Ch. 3 self-checks 16-19, ex. 14-15; Ch. 4 self-checks 7-9, 11-16, ex. 8-10)*

6. Exercise 4.9, p315 ("evenSumMax"). Write code to prompt the user for integers and print the total even sum and the maximum of the even numbers typed. You may assume that the user types at least one non-negative even integer.

```
how many integers? 4
next integer? 2
next integer? 9
next integer? 18
next integer? 4
even sum = 24
even max = 18
```

CSE 142 Section Handout #4
Style Sheet

Consider the following program:

```
import java.util.*;

public class Sect4 {
   public static void main(String[] args) {
       double hours = 0.0;
       checkSleep(hours);
       System.out.println();
       printFriends();
   }

   public static void checkSleep(double hours) {
       Scanner console = new Scanner(System.in);
       System.out.print("How many hours of sleep do you get? ");
       hours = console.nextDouble();
       if (hours < 7) {
           System.out.println("You're getting too little sleep.");
       }
       if (hours >= 7 && hours <= 9) {
           System.out.println("You're getting the recommended amount of sleep.");
       }
       if (hours > 9) {
           System.out.println("You're getting more sleep than is recommended.");
       }
   }

    public static void printFriends() {
       Scanner console = new Scanner(System.in);
       System.out.print("How many friends do you have? ");
       double friends = console.nextDouble();
       if (friends < 50) {
           System.out.println("You are friends with " + friends / 74000000.0 +
               " percent of the world.");
           System.out.println("You need to get more friends!");
       } else if (friends < 250) {
           System.out.println("You are friends with " + friends / 74000000.0 +
               " percent of the world.");
           System.out.println("You have an average number of friends.");
       } else if (friends >= 250) {
           System.out.println("You are friends with " + friends / 74000000.0 +
               " percent of the world.");
           System.out.println("Whoa there!  You have a lot of friends.");
       }
   }
}
```

While this method would receive full external correctness by producing the desired output, it would not receive full internal correctness. List all style issues you can find.