Homework1

Note: Submit your work (upload the .java source code files ONLY, not the compiled .class files!) through the "Homework1" link on Brightspace. You may submit an unlimited number of times; we will only grade the last/latest submission attempt, but be sure to **attach all of your files to each submission attempt**. Be sure to include your name and Stony Brook ID number in a comment at the beginning of each file that you submit.

Due: Tuesday, September 24, 11:59pm **Total: 40 points (5 points per problem)**

Submission Instructions: Name your java classes for this assignment as:

Problem1: Schedule.java, Problem2: ShowMeTheMoney.java,
Problem3: CarFuelConsumption.java, Problem4: IsLeapYear.java,
Problem5: Eggs.java, Problem6: WindChill.java,
Problem7: Divisible.java, Problem8: Triangle.java

1. (Schedule.java) Write a Java program Schedule that prints your name, SBU ID number and weekly class schedule. It should have the general appearance of the example below with your name on the first line, followed by your SBU ID # on the second line, followed by your weekly schedule. Adjust the time slots as necessary to correctly reflect your actual class schedule. If you are taking fewer than four courses, add fictional courses to bring the total up to four.

```
John Smith
1234567890
Time              Monday            Tuesday           Wednesday                  Thursday          Friday
8:30am-9:50am                       EGL101                                       EGL101
10:00am-11:20am   CSE114                               CSE114
11:30am-12:50pm
1:00pm-2:20pm                       MAT125                                       MAT125
2:30pm-3:23pm     FRN111                                FRN111                                     FRN111
4:00pm-5:20pm
5:30pm-6:50pm                       PHY127                                       PHY127
```

2. (ShowMeTheMoney.java) Before 1961, people living in Great Britain had to deal with a pretty complicated currency consisting of pounds sterling, shillings, pence, farthings, and other coins. The currency system worked like this:

   - 1 pound sterling = 20 shillings
   - 1 shilling = 12 pence ("pence" is the plural of "penny")
   - 1 penny = 4 farthings

Write a program (ShowMeTheMoney.java) that reads in a single integer, the number of farthings, and then prints a sequence of integers that represents the fewest number of each bill or coin needed to represent that amount, starting with the largest denomination. (Hint: use the % operator.)

For example, if the argument is 2563 farthings, then the program prints 2, 13, 4, 3 because 2,563 farthings is equivalent to 2 pounds, 13 shillings, 4 pence and 3 farthings.

Examples:
(You don’t have to print the currency each number is associated with, just printing the number is fine.)

```
Enter in the number of farthings: 2563
Sterling 2
Shilling 13
Penny 4
Farthings 3
```

```
Enter in the number of farthings: 4900
Sterling 5
Shilling 2
Penny 1
Farthings 0
```

```
Enter in the number of farthings: 762
Sterling 0
Shilling 15
Penny 10
Farthings 2
```

3. (CarFuelConsumption.java) Write a program (CarFuelConsumption.java) that reads the following arguments, in this order:
    - minutes: the number of minutes a car is driven on a long track
    - mph: the constant speed of the car, given in miles per hour
    - mpg: the fuel efficiency of the car, given in miles per gallon
      The function computes and prints the number of gallons of gasoline consumed by a car driven down a long track over the given time period. Use floating-point division only.

Examples

```
Input the number of minutes a car is driven on a long track: 75
Input the constant speed of the car in miles per hour: 35.2
Input the fuel efficiency of the car in miles per gallon: 20.6
2.1359223300970873
```

```
Input the number of minutes a car is driven on a long track: 60
Input the constant speed of the car in miles per hour: 60
Input the fuel efficiency of the car in miles per gallon: 30
2.0
```

```
Input the number of minutes a car is driven on a long track: 164.3
Input the constant speed of the car in miles per hour: 45.2
Input the fuel efficiency of the car in miles per gallon: 19.41
6.376747381075049
```

Hint: minutes/60 * mph * 1/mpg

4. (IsLeapYear.java) A leap year is a calendar year which contains an additional day in the month of February. The purpose of the additional day every four years is to keep our calendar in sync with the Earth's revolution around the sun. A leap year can be determined by the following conditions:

   If the year is divisible by 400 then it is a leap year
   If the year is not divisible by 400 but divisible by 100 then it is not a leap year
   If the year is not divisible by 400, not divisible by 100 but divisible by 4 then it is a leap year
   If it fails all the condition above then the year is not a leap year

   Write a program (IsLeapYear.java) which prompts the user to enter in a year and determine whether or not it is a leap year or not.

   Examples
   Enter in a year: 2021
   2021 is not a leap year

   Enter in a year: 2020
   2020 is a leap year

   Enter in a year: 1400
   1400 is not a leap year

   Enter in a year: 1900
   1900 is not a leap year

5. (Eggs.java) Stony Brook Dairy Farm sells organic brown eggs to local customers. It charges $3.25 for a dozen eggs, or 45 cents for individual eggs that are not part of a dozen. Write a program that prompts a user for the number of eggs in the order and then display the amount owed with a full explanation. For example, typical output might be, ***You ordered 27 eggs***. ***That's 2 dozen at $3.25 per dozen and 3 loose eggs at 45 cents for a total of $7.85***. Save the program as Eggs.java.

6. (WindChill.java) How cold is it outside? The temperature alone is not enough to provide the answer. Other factors including wind speed, relative humidity, and sunshine play important roles in determining coldness outside. In 2001, the National Weather Services (NWS) implemented the new wind-chill temperature to measure the coldness using the temperature and wind speed. The formula is

$$t_{wc} = 35.74 + 0.621t_a - 35.75v^{0.16} + 0.4275t_a v^{0.16}$$

   Where t_a is the outside temperature is measured in degree Fahrenheit, v is the speed measured in miles per hour, and t_{wc} is the wind-chill temperature. The formula cannot be used for wind speed below 2 mph or temperature below -58 Fahrenheit or above 41 Fahrenheit.

Write a program that prompts the user to enter a temperature between -58 Fahrenheit and 41 Fahrenheit and a wind speed greater than or equal to 2 then displays the wind-chill temperature. Use Math.pow(a, b) to compute $v^{0.16}$.

7. (Divisible.java) Write a program that prompts the user to enter an integer and determine whether it is divisible by 5 and 6, whether it is divisible by 5 or 6, and whether it is divisible by 5 or 6, but not both. Here is a sample run of the program:

```
Enter an integer: 10
Is 10 divisible by 5 and 6? false
Is 10 divisible by 5 or 6? true
Is 10 divisible by 5 or 6, but not both? true
```

8. (Triangle.java) Write a program that reads three edges (sides) for a triangle and classify it as Equilateral, Scalene, Isosceles if the input is valid. The input is valid if the sum of every pair of two edges is greater than the remaining edge. You may assume that the user inputs a positive value for the edges.

Here is a sample run.

```
Side1:3
Side2:3
Side3:9
Output: Invalid input!


Side1:3
Side2:3
Side3:3
Output: Equilateral

Side1:10
Side2:10
Side3:1

          Output: Isosceles

Side1:3
Side2:4
Side3:5

          Output: Scalene
```

[**Hint:** Equilateral: All sides equal, Isosceles: Two sides equal, Scalene: All sides different]