Faculty of Arts & Sciences
Department of Computer Science
CMPS 200—Introduction to Programming
Exam 2 (90 minutes)

Problem 1 `Triangle.java` (25%)

A triangle is ***valid*** if the sum of any two sides is greater than the third side. Valid triangles can be classified according to the relative lengths of their sides:

- In an **equilateral** triangle all sides have the same length.
- In an **isosceles** triangle, two sides are equal in length.
- In a **scalene** triangle, all sides are unequal.

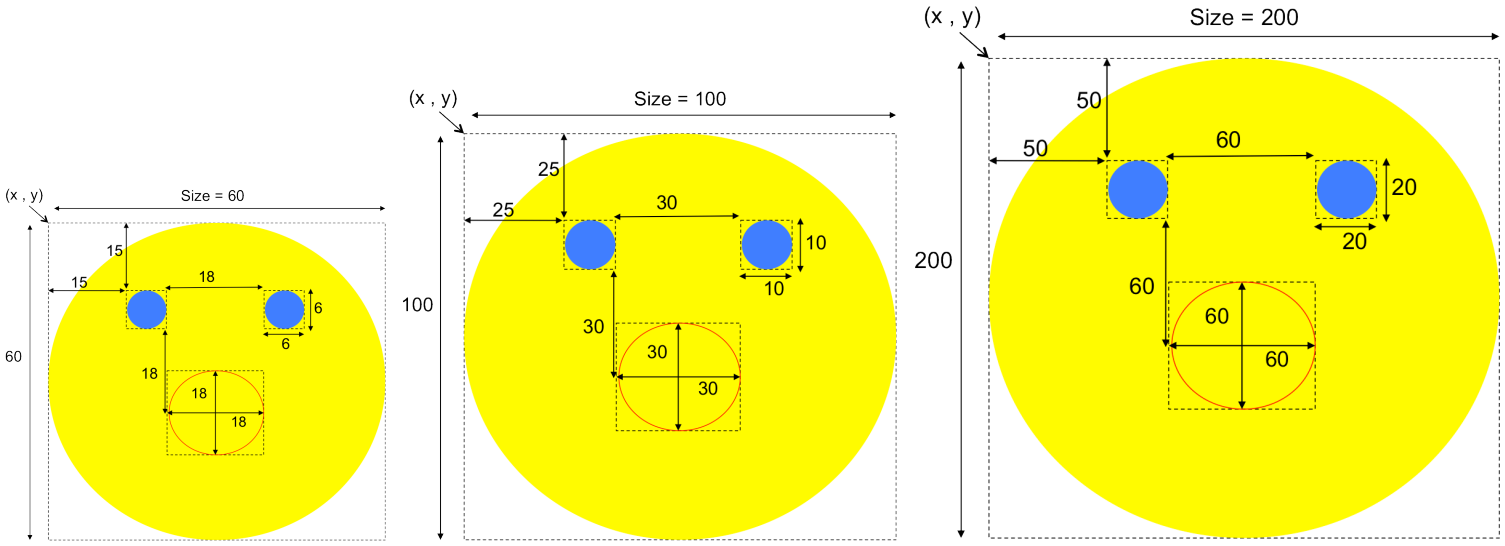
Write a program `Triangle.java` that takes the length of three sides as command line and prints "equilateral", "isosceles" or "scalene" describing the triangle type. If the input doesn't create a valid triangle your program must print "invalid".

Example:

```
> java Triangle 3.7 4.2 2.5
Scalene.

> java Triangle 3.7 3.7 2.5
Isosceles.

> java Triangle 2.7 5.2 1.3
Invalid.

> java Triangle 3.7 3.7 3.7
Equilateral.
```

Problem 2 `Face.java` (35%)

Write a method `drawFace` that takes as arguments the *coordinates (x, y)* as integers, the integer *size,* and the string *mood,* and draws a (*Happy*, *Sad* or *Fine*) smiley face based on the following scale reference:



The face should be in color yellow, the eyes in color blue, and the mouth in color red. The width and height of the drawing panel could be of any size, as long as it is larger than the face's size. **The mood is case sensitive**.

Write a program `Face.java` that uses the method `drawFace` to draw a smiley face based on the *coordinates (x, y), size* and *mood* entered by the user.

Example:

```
> java Face
Enter the x coordinate of the face: 30
Enter the y coordinate of the face: 30
Enter the size of the face: 100
Are you Happy, Sad, or Fine? Sad
```

```
> java Face
Enter the x coordinate of the face: 40
Enter the y coordinate of the face: 40
Enter the size of the face: 200
Are you Happy, Sad, or Fine? Fine
```

```
> java Face
Enter the x coordinate of the face: 30
Enter the y coordinate of the face: 30
Enter the size of the face: 60
Are you Happy, Sad, or Fine? Happy
```

```
> java Face
Enter the x coordinate of the face: 30
Enter the y coordinate of the face: 30
Enter the size of the face: 60
Are you Happy, Sad, or Fine? I hate CMPS 200
Unknown Mood!
```

Problem 3 `LuckPercentage.java (40%)`

An integer greater than one is called a prime number if its only positive divisors (factors) are one and itself. For example, the divisors of 7 are 1 and 7; and the first six primes are 2, 3, 5, 7, 11 and 13.

In this problem, we introduce a prime character, which is a character with the corresponding ASCII code translating to a prime number. For example, the ASCII code of the character 'A' is 65, thus, 'A' is not a prime character. The ASCII code of the character 'C' is 67, therefore, 'C' is a prime character.

In this problem, your task is to define the ***percentage of luck*** for each name provided by the user. Names are constituted by characters with the ASCII code from 65 ('A') to 122 ('z'), including underscores.

We compute the percentage of lucky names by calculating the average weight of the characters constituting each name, where each character is given either the weight 100 if it is prime or the weight 0 otherwise. Given a name N of length L, the percentage of luck for N is computed as follow:

$$W(N) = \frac{\sum_{i=0}^{L-1} w_i}{L}$$ where, w_i is the weight of character at index i.

Write a program, `LuckPercentage.java`, which accepts from the command line a word and then calculates the percentage of luck associated with that name.

Example:

```
> java LuckPercentage SISCO
W("SISCO") = 100/5 + 100/5 + 100/5 + 100/5 + 100/5 = 100
Thus, the word SISCO is 100% Lucky
> java LuckPercentage cake
W("cake") = 0/4 + 100/4 + 100/4 + 100/4 = 75
Thus, the word cake is 75% Lucky
```

Notes and Submission Instructions

- Your exam submission must consist of a single zip archive named **`s#_exam1_`*`netid`** that contains your properly commented three source files (.java files) only (**Triangle.java**, **`Face.java`**, **`LuckPercentage.java`**). No other files will be accepted. We will compile and run your programs.
- You should write the appropriate static methods to show structure in your solution and to help you in writing other methods.
- Give meaningful names to methods and variables in your code.