AP CSP Processing Final Project Report

1. The program I created was a game. You're Joe the ninja, and you're hungry! Steal all the food as quickly as possible from the table before being caught. You get caught once the light turns on. There will only be a 0.33 second "warning" before the light turns on, so you have to hide real quickly!

   You can move left/right with the left/right arrows. You can steal food by pressing the up arrow. And you can hide using the down arrow.

   Some of the features of this program include the random creation of food, the movement of the food corresponding to the ninja's movement, movement of the ninja with the down key, animation of the fruit as they are eaten, and various aspects of a game (keeping score, time, lives).
2. An example of each of the following elements:
   - Use of basic primitive shapes: Ellipses are used for the foods and ninja's head; lines are used for the ninja's body; a rectangle is used for the table; and triangles are used in the apples.
   - Diverse and intentional use of color: The default stroke is all black, but the ninja's stroke is made partially transparent when hiding; the oranges are orange; the apples are red with a green leaf; the table is brown; the background is dark when the ninja is free to move, and light when he is caught (and in the middle as a warning)
   - Some element of randomization: the fruits are randomly placed along the table; the intervals of time before the light turns on again are randomized
   - Looping structure: looping over the foods to draw them all
   - Selection (if) statement: only printing a food if it hasn't been eaten
   - Animation!: The table and foods are animated when the person moves; the foods are animated when eaten
   - Any element of user interaction: the four arrow keys are used to play the game
   - Use at least one variable: example variables include personXPos, numLives
3. I learned a lot of the convenience/shorthand functions and constants provided by Processing, such as PI, cos(), sin(), millis(), etc. I used a lot of help from the Processing documentation.
4. I would add more interesting foods, as well as different point values for the different foods and different difficulties for picking them up. If I had more time past that, I would create multiple levels with harder difficulties (shorter warning times).
5. I'd give myself a 5. I didn't copy any code, and I spent all of the class time given for this project working on the project. I also spent a lot of time writing multiple drafts of the project before I settled on a final version: my first idea was to create a beach scene using mathematics (e.g., the fibonacci sequence for shells) to create cool graphics, but it didn't look as good as I wanted to, and I ended up writing two drafts for this game because the first one was messy and I wanted to reorganize my code using OOP.