

## CPS109 Lab 7

Most of the questions in this lab come from Chapter 6 of the course text, Introduction to Computing and Programming in Python, by Guzdial and Ericson. Please put your answers (numbered) in a document and submit it on D2L as a PDF file. Other formats are not accepted.

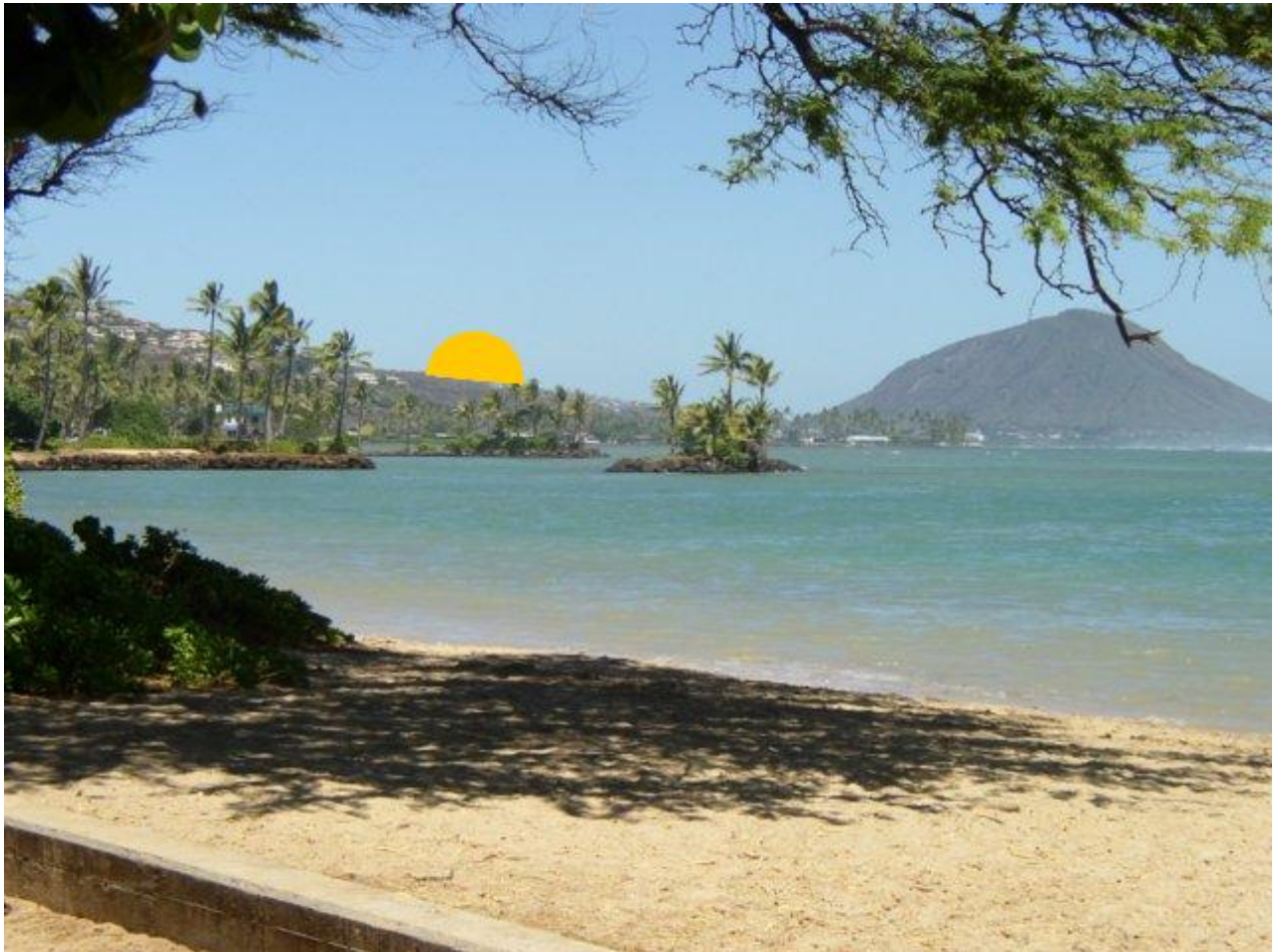
The **learning objectives** for Chapter 6 include:

- to mirror pictures
- to copy pictures, to make a collage, to scale and rotate
- to use nested loops
- to loop through just part of a two-dimensional array
- to use print statements to help in debugging

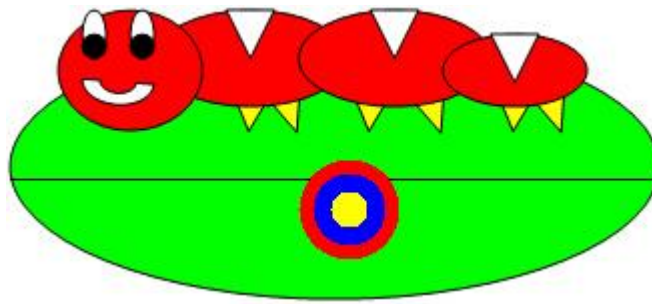
### To do:

For each question, include in your solution document your code and your example picture.

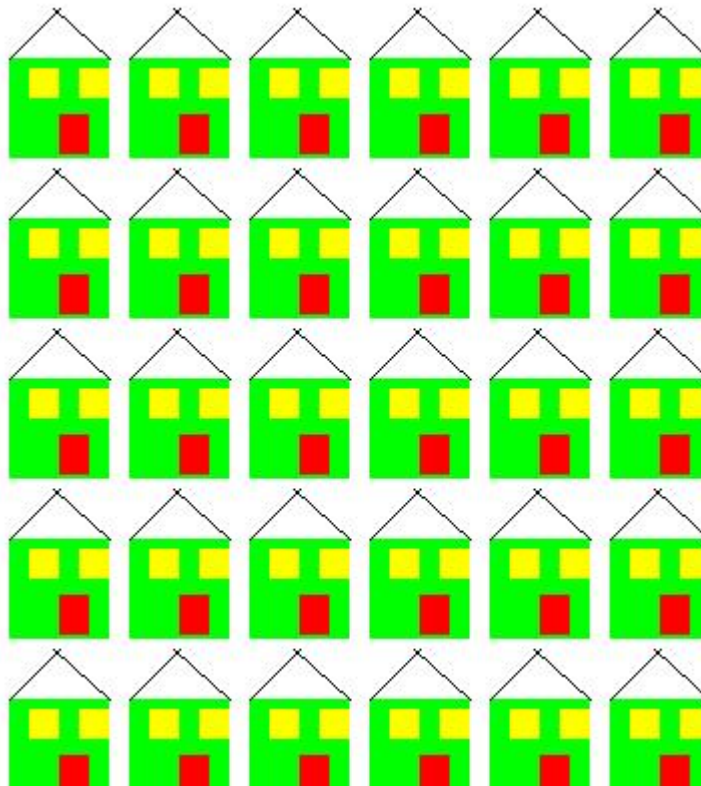
- 1) Write a function **addSunset()**, which puts an orange sun-like hemisphere on the beach picture, as shown below. Use the function `addArcFilled(picture, x1, y1, width, height, startangle, degrees, color)`, which you can read about in JES under the tab JES Functions > Pictures.



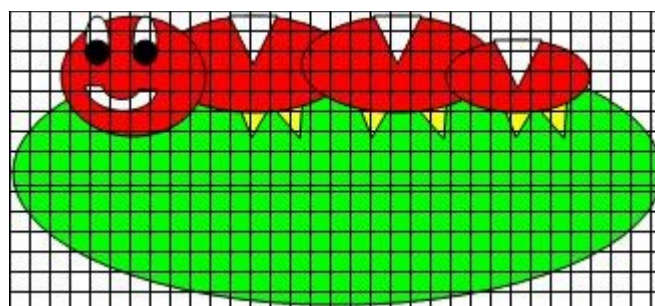
- 2) Write a function **addBullseye(picture, x1, y1, width)** which uses the JES drawing functions to draw a bullseye on a picture at `x1, y1`, with the given diameter called `width`. Here is an example application to the caterpillar.



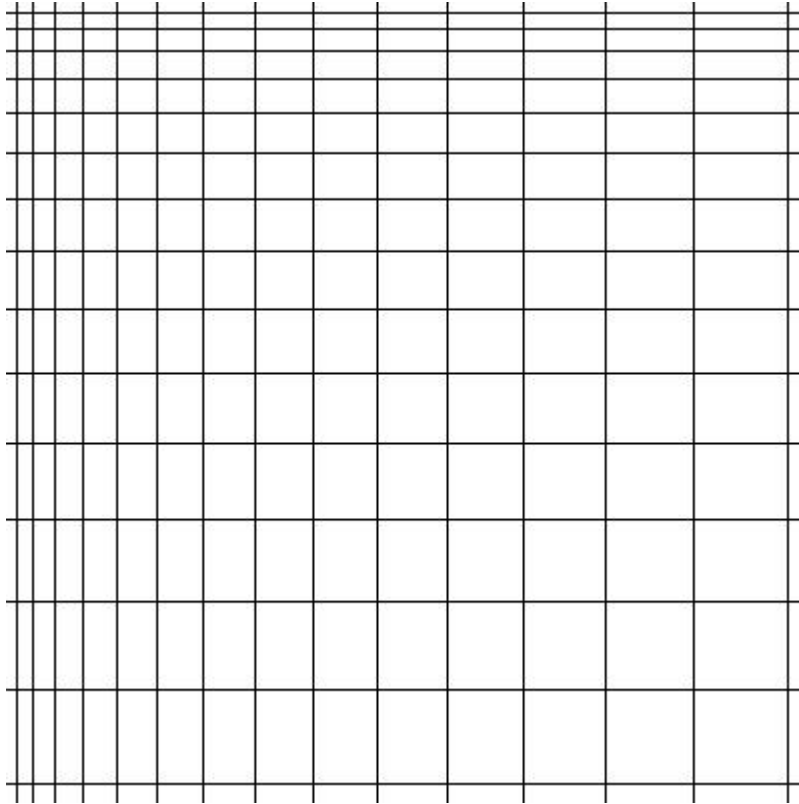
- 3) Write a function **addHouse(picture, x1, y1)**, which draws a simple child's version of a house (a door, two windows, two walls and a roof) at position x1, y1. Use a loop to spread a bunch of houses around on a canvas.



- 4) Write a function **drawGrid(picture, spacing)**, which uses the JES addLine() function to draw horizontal and vertical lines over a picture making a grid with a separation between the lines of <spacing>.



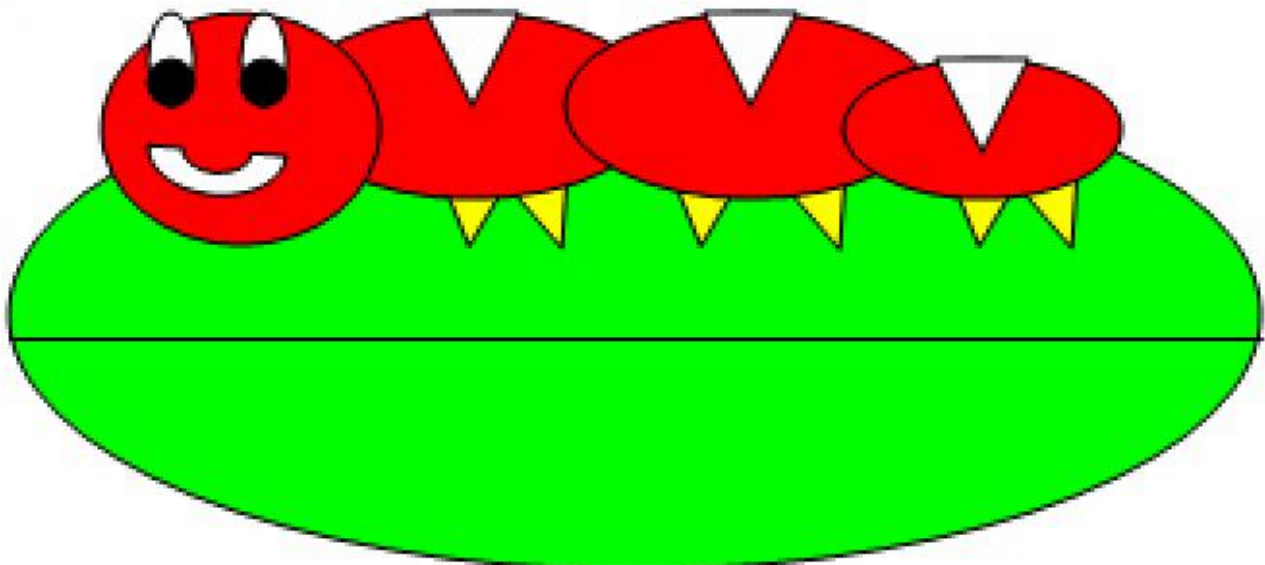
- 5) Write a function **drawGrid2(picture)** which draws horizontal and vertical lines over the picture using the JES `addLine()` function. Let the spacing between the horizontal lines start at 5, and gradually increase with each new line: 5, 8, 11, 14, 17, ... Do the same gradual increase for the vertical line spacing. As always, show your code and example picture.



- 6) Write a function to **flipHorizontally(picture)** a picture over, so that if the llama or whatever is looking right, then it ends up looking in the opposite direction. Show code and example application.

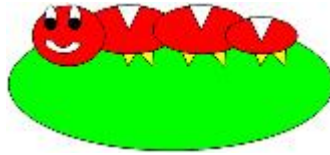


- 7) Write a general **scaleup(picture)** function which takes in any picture and creates and returns a new picture twice as big, but using the trick of increasing the sourceX and sourceY by 0.5, to repeat pixels in both the x- and y- directions while increasing targetX and targetY by 1. Use the function **makeEmptyPicture(width, height)** to create a canvas for the picture.





- 8) Write a general **scaledown(picture)** function which takes in any picture and creates and returns a new picture twice as small by using the trick of copying every other pixel to the canvas. Use the **makeEmptyPicture(width, height)** function for creating the canvas.



- 9) Write a function **mirror20(picture)** that mirrors the leftmost 20 pixels to pixels 20 to 40 in the x-direction. The rest of the picture stays the same.

