Specification for Assignment 07

*Your submission for this assignment **must include your full name** and your nine-digit **student number as a comment** at the top of the **source file you submit**. All source code files must be written using the **Python 3 programming language** and must run on the course's **official virtual machine**.*

***Submissions that crash** (i.e., terminate with an error) on execution will **receive a mark of 0**.*

Officially, the Due Date for this Assignment is:

***Friday, November 5th, 2021**, at **11:59pm EST**.*

Late Submissions are Accepted Without Penalty Until Sunday, November 7th, by 11:59pm EST.
Submissions received after that will not be accepted and will receive a mark of 0.

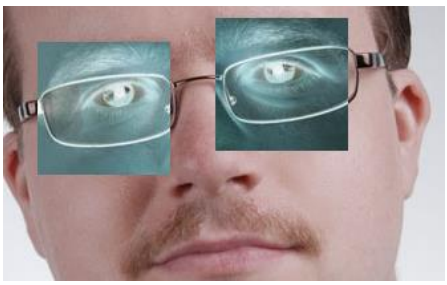
To further practice with nested looping control structures, for this assignment you will design and implement a program that performs a very elementary photomanipulation.

In order to complete this task, you will need to:

- read about pygame's "image.load"[1], "mouse.get_pressed", and "mouse.get_pos" functions[2]
- read about the pygame Surface methods "get_size", "blit", "get_at", and "set_at"[3]
- choose whether you would like to implement a simple "negation" effect (below left)…
  …or if you would prefer something more impressive, like "blur", "sharpen", or "edge detect"[4]

Your submission for this assignment:

- must be a source code file with filename 'comp1405_f21_#########_assignment_07.py'
- must initialize pygame, load an image, and use get_size to find the loaded image's dimensions
- must create a window that is the correct size and then "blit" the loaded image
- must use the loop (below right) to repeatedly get input, apply the effect, and update the display
- must permit the user to use the mouse to specify a region of the image (i.e., by clicking)
- must use nested loops, "get_at", and "set_at" to apply an effect to the region the user selected

the difference between the maximum colour component value (i.e., 255) and the red, green, and blue component values of a colour, can be used to get the the negated (or inverted) colour
 	
e.g.,
(200, 100, 50)
becomes
(55, 155, 205)
		`exit_flag = False`
		`while not exit_flag:`
		
		`  # loop body goes here`
		
		`  for e in pygame.event.get():`
		`  if e.type == pygame.QUIT:`
		`  exit_flag = True`

[1] At the time of creating this document, this function is detailed at https://www.pygame.org/docs/ref/image.html.
[2] At the time of creating this document, these functions are detailed at https://www.pygame.org/docs/ref/mouse.html.
[3] At the time of creating this document, these methods are detailed at https://www.pygame.org/docs/ref/surface.html.
[4] More complex effects can be accomplished using "kernels": https://en.wikipedia.org/wiki/Kernel_(image_processing)