

Answer the questions in the spaces provided on the question sheets. If you run out of room for an answer, continue on the back of the page.

Name: _____

Functions

1. Given the functions

$$F(m, a) = m \cdot a$$

and

$$a(t) = t^2 + 4t$$

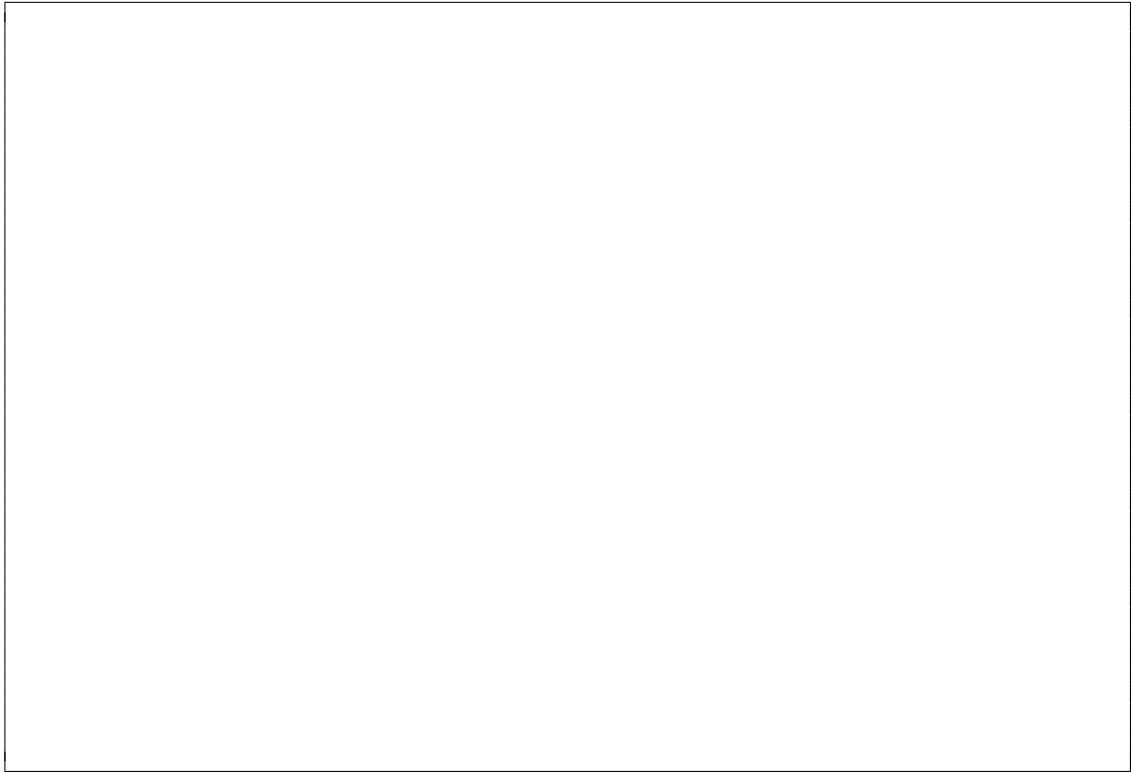
compute the following:

- 1 (a) $F(3, 5)$

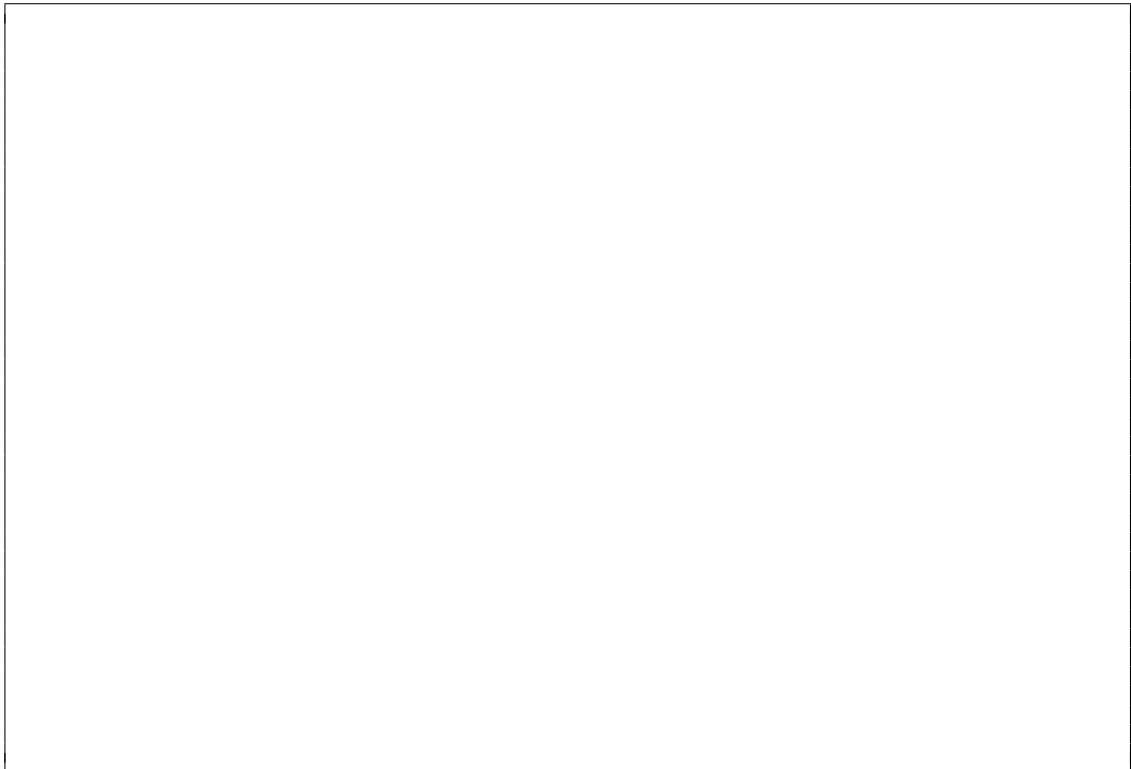
- 1 (b) $a(3)$

- 3 (c) $F(3, a(3))$

- 5 (d) Re-write $F(m, a)$ as a Python function.



- 5 (e) Re-write $a(t)$ as a Python function.

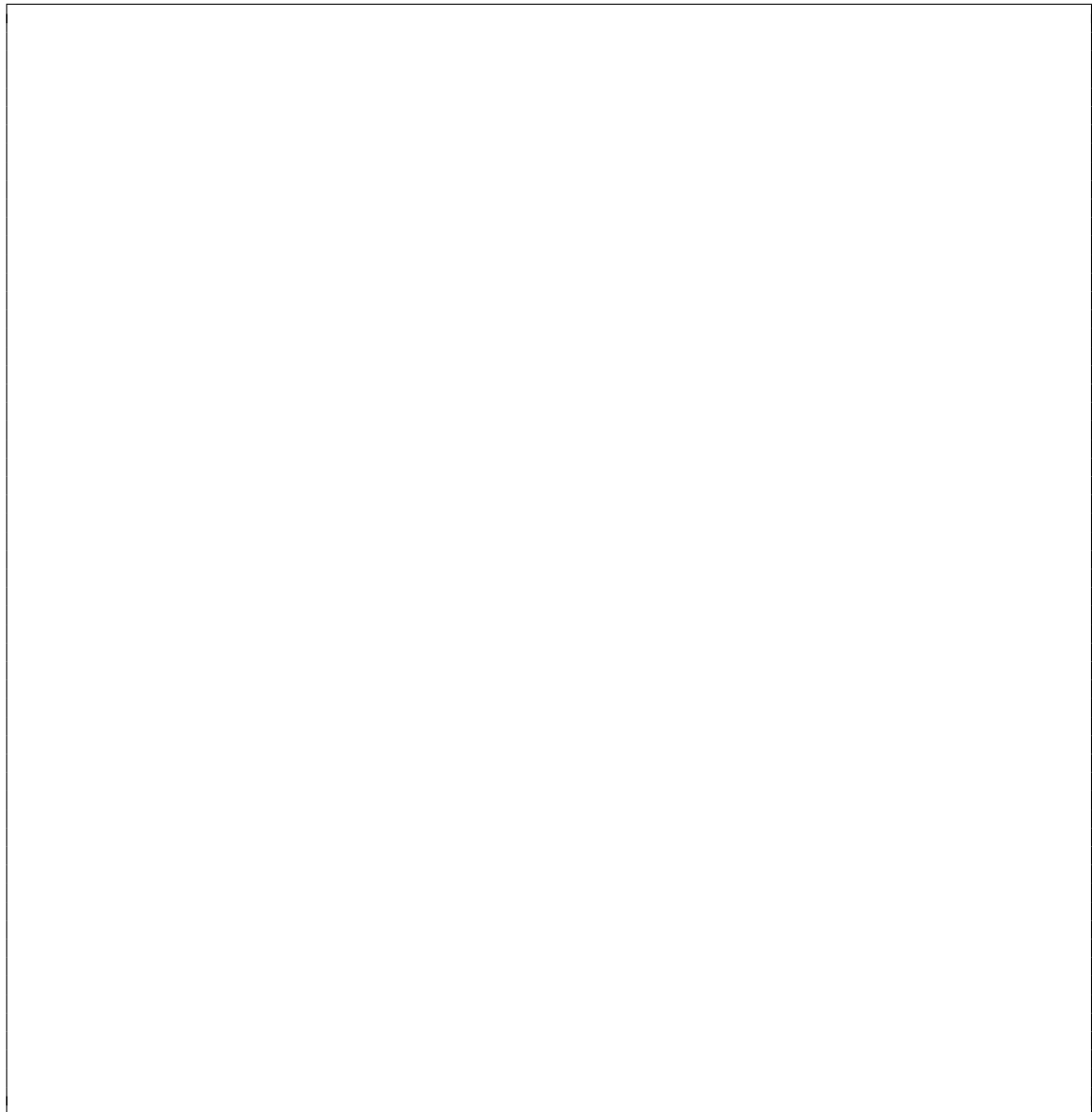


- 5 (f) Re-write $F(m, a)$ so that the default acceleration is 5. That is, one should be able to call this function using $F(3)$.

2. Colors are often given as a tuple of RGB values: (R, G, B), where (0, 0, 0) is black and (255, 255, 255) is white. The brightness of a color can be adjusted by scaling each RGB component by the same amount.

- 5 (a) Scale the color (100, 200, 100) by 125% (1.25). The resulting color will then be:

- 10 (b) Write a function called `adjust_brightness` that takes in a color tuple *color* and scales it by a factor *factor*. Don't forget about clipping issues!



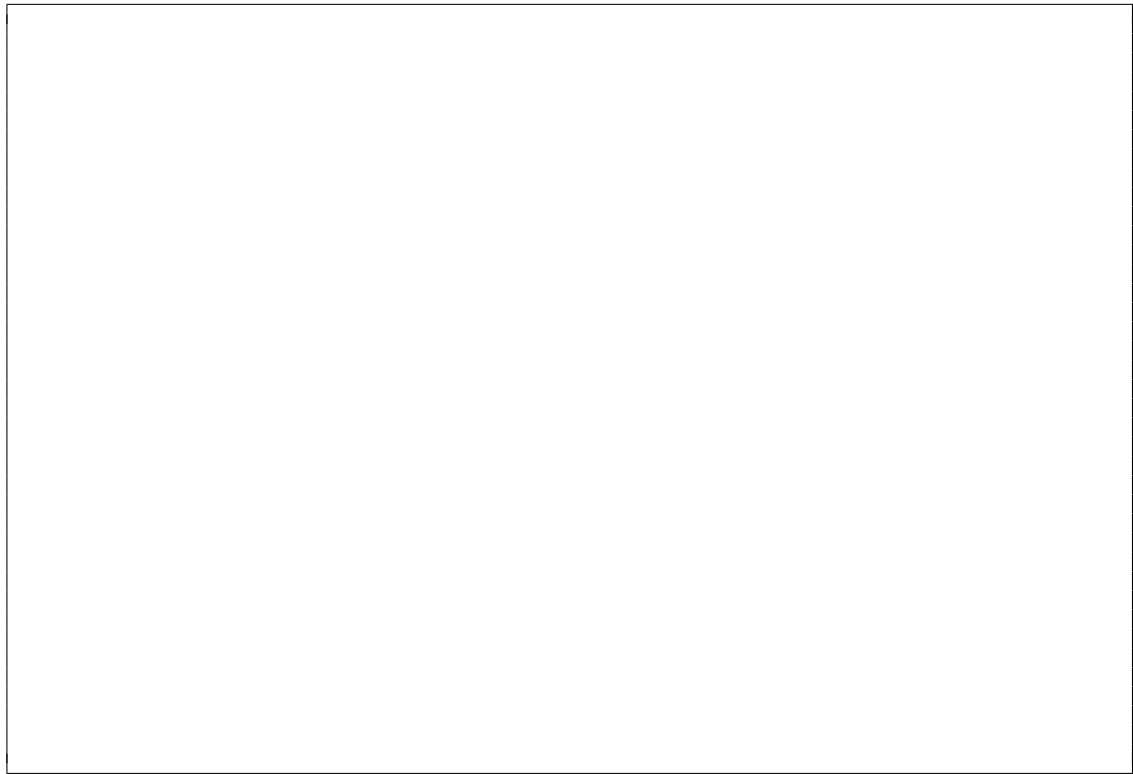
Classes

Classes are Python's way of defining objects¹. A class is a *template* (or a blueprint) of an object. Unlike other languages, `self` is explicitly passed as the first argument to instance methods.

3. In this section, we will work with the `PredatorTank`, which is a variation of the `Tank` class as discussed in the text.

- 10 (a) Create a class called `PredatorTank`. It should have an initialization method (`__init__`) that takes in a `name`, `health`, `attack`, `position`, and `range`.

¹<http://docs.python.org/tutorial/classes.html>

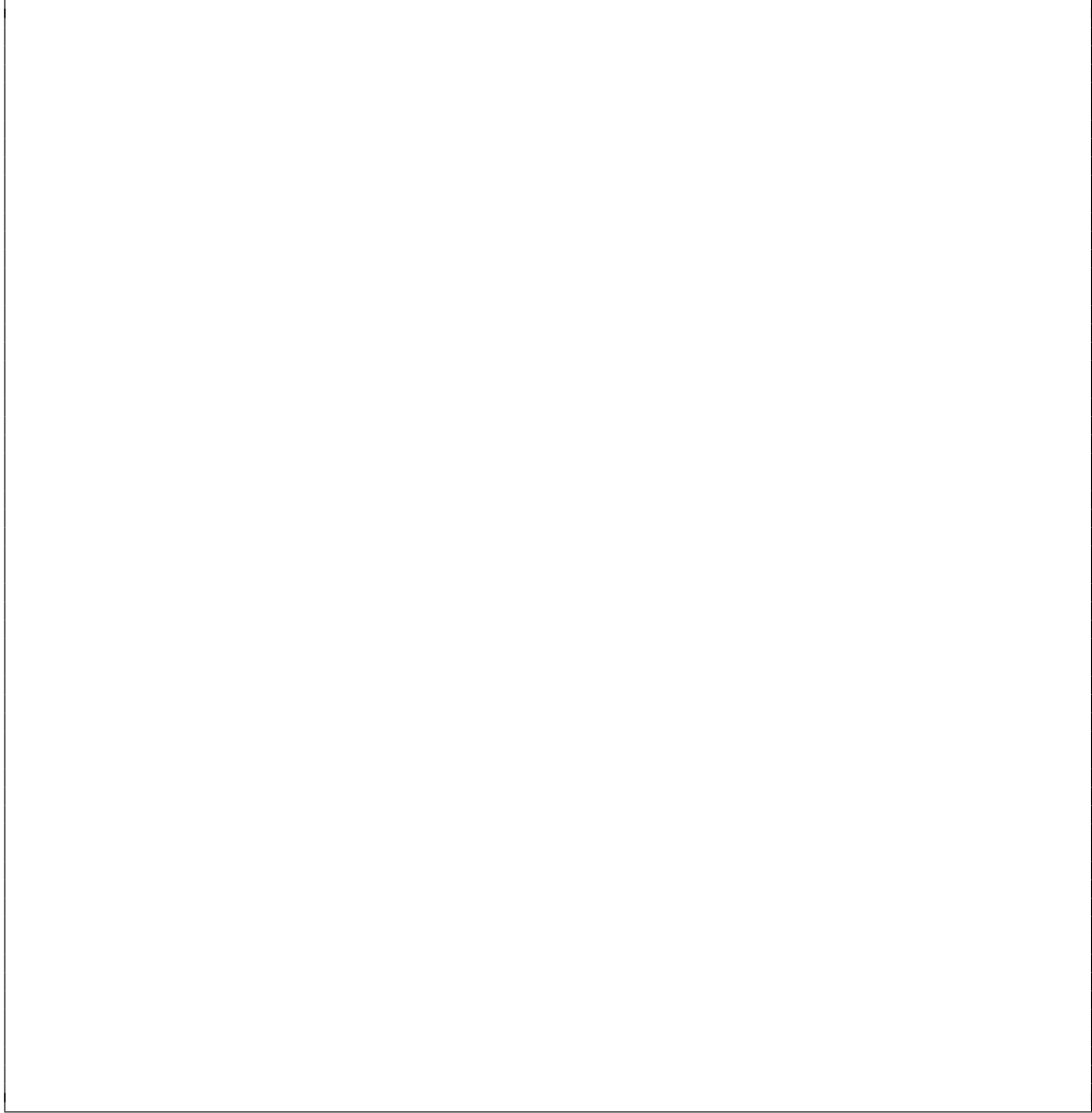


- (b) Create three tanks: `t1`, `t2`, and `t3`. You can name these tanks “T1”, “T2”, and “T3”. Each tank will have a health of 100, an attack of 5, and a range of 7. The position of T1 is (0, 0). The position of T2 is (10, 15). The position of T3 is (11, 16).



- (c) Plot the tanks on a graph. Label your axes.
- (d) Using the distance formula, compute the distance between T1 and T2 (you can use `math.sqrt` within Python).
- (e) Within this class, write a method called `fire(self, target)`, that fires upon another `PredatorTank`. It should subtract `attack` units from the target, but only if it's own health is greater than 20 (otherwise, the unit is too damaged to fire). If the unit is out of range, then the attack will not be successful (i.e, do nothing).

The health of a unit can never fall below zero.



- 5 (f) Draw a UML class diagram of the completed `PredatorTank`.

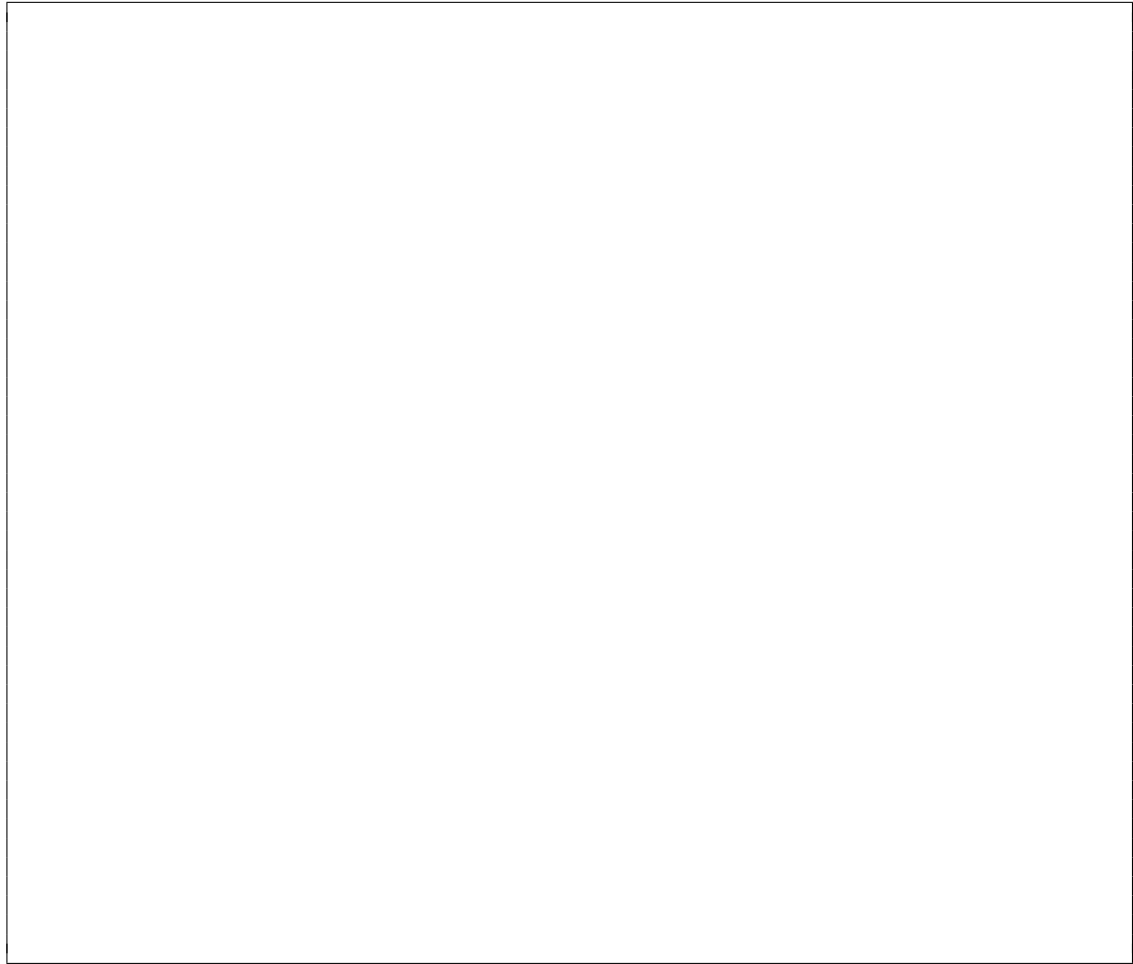


Game Loop

4. These questions relate to Pygame and the game loop. Use a resolution of 640x480.

10

(a) Write a minimal game loop, using Listing 3.3 as a starting point.



- 5 (b) Load the **Character Boy** sprite using `pygame.image.load`² and store it in the variable `boy`. This code should appear before the game loop, since the **Surface** has to be loaded only once.

- 5 (c) Fill the background so that it is black.

- 5 (d) Blit the `boy` to the screen at position (50, 50).

²<http://www.pygame.org/docs/ref/image.html#pygame.image.load>

- 10 (e) Modify the code so that the avatar moves one pixel to the right at each iteration of the game loop. If the unit goes past the screen, reset it so that it wraps around.

- 5 (f) What happens if you forget to fill the screen? Why does this happen?

5. In the previous questions, you automatically moved the avatar one pixel the right at every iteration of the game loop. In this question, you will control movement through keyboard controls. For all cases, ensure that the avatar “wraps around” the screen.

Use the file `game.py` to add your programming logic.

- 10 (a) Add logic to the game loop so that pressing the right arrow moves the character to the right (`KEYDOWN`). The avatar should keep moving right until the key is released (`KEYUP`).

- 10 (b) Do the same for the left arrow, moving the avatar one pixel to the left.

- 15 (c) Add a second avatar, `Character Girl`, to the screen at location (300, 300). Use `A` and `D` to control this character.

You now have a mechanism to test your AI! You can use your arrow keys to move your avatar and ultimately see how an AI opponent will respond.

Question	Points	Score
1	20	
2	15	
3	15	
4	40	
5	35	
Total:	125	