National University of Singapore
School of Computing
CS1010S: Programming Methodology
Semester I, 2022/2023

# Side Quest 5.1
# Circle Manipulation

Release date: 7th September 2022
**Due: 13th September 2022, 23:59**

## Required Files

- sidequest05.1-template.py
- hi_graph.py

## Information:

For your convenience, the template file **`sidequest05.1-template.py`** contains a line to load the Python source file **`hi_graph.py`**. Use the template file to answer the questions.

This side quest consists of **two** tasks.

## Task 1: (2 marks)

**Note: For each part, please state your answer as numbers instead of strings!**

Execute the following code:

```
draw_connected(200, unit_circle)
```

Then execute the following:

```
draw_connected(200, alternative_unit_circle)
```

Can you see a difference? Now try using `draw_points` instead of `draw_connected`. Also, try using other drawing functions to draw `unit_circle` and `alternative_unit_circle`.

After observing, try to answer these questions:

(a) Suppose we have

```
tupify_point(unit_circle(some_t)) ==
            tupify_point(alternative_unit_circle(0.5))
```

What is the value of `some_t`?

(b) Suppose we have

```
tupify_point(unit_circle(0.09)) ==
            tupify_point(alternative_unit_circle(other_t))
```

What is the value of `other_t`?

(c) Give an example of two values `t1` and `t2` not included in the previous two questions such that

```
tupify_point(unit_circle(t1)) ==
            tupify_point(alternative_unit_circle(t2))
```

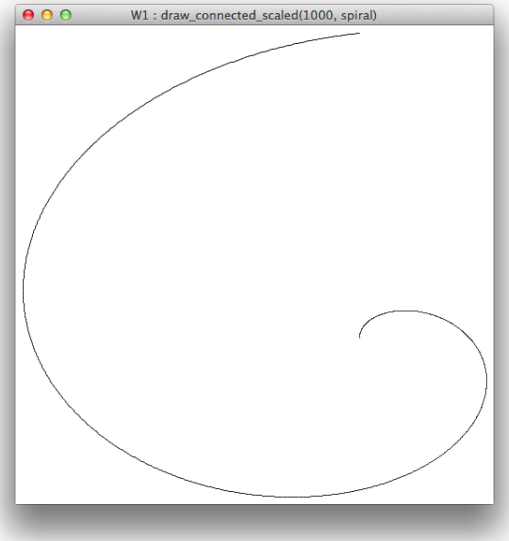
(d) Again, give another example of two values `other_t1` and `other_t2` not included in the previous three questions such that

```
tupify_point(unit_circle(other_t2)) ==
            tupify_point(alternative_unit_circle(other_t1))
```

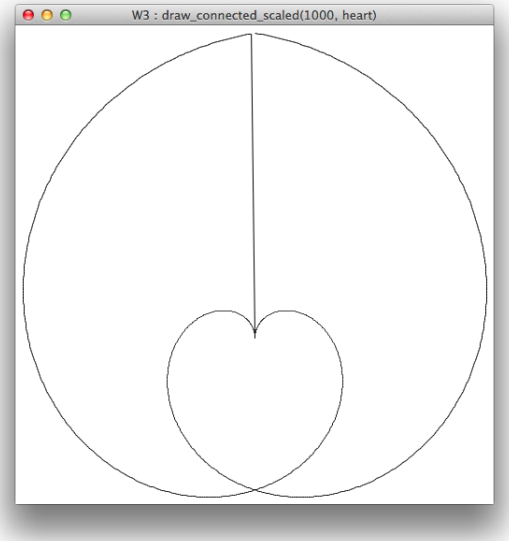
## Task 2: (3 marks)

(a) Using the definition of the `unit_circle` as a reference, define a new curve function `spiral` that draws a 'circle' which mimics a spiral.

(b) Define a new curve function `heart` that draws a curve by connecting 2 spirals. You should make use of your `spiral` function from the previous part to produce the curve.




draw_connected_scaled(1000, spiral)




draw_connected_scaled(1000, heart)