

Functions, Part One

Demo 1

What does this code do?

```
def secret():  
    line(40, 30, 75, 60)  
    line(75, 30, 40, 60)  
  
size(300, 300)  
secret()
```

Answer: The program defines a function that draws an X (using two lines) at a fixed location. The program then calls that function to draw the X on the canvas

Solution

This code can be found in: `cmpt140-ch04-py/cmpt140_ch04_lit_1/cmpt140_ch04_lit_1.pyde`

Exercise 1

```
def draw_moon():  
    ellipse(250,250,300,300)  
  
def draw_crater(x,y):  
    ellipse(x,y,30,30)  
  
def draw_flag():  
    line(205,205,205,295)  
    rect(205,205,50,30)
```

Write code that **calls** the functions above to:

- (a) Draw the moon with a flag planted on its surface

Solution

```
# CMPT 140 - Functions, Part One
# Topic: Function Calls
# Part (a)

size(500,500)

def draw_moon():
    """ draws moon of diameter 300px in centre of screen """
    ellipse(250,250,300,300)

def draw_crater(x,y):
    """
    draws crater of radius 30px with (x,y) centre coordinate
    x: x-coordinate of crater centre
    y: y-coordinate of crate centre
    """
    ellipse(x,y,30,30)

def draw_flag():
    """ draws flag planted at (205,295) """
    line(205,205,205,295)
    rect(205,205,50,30)

draw_moon()
draw_flag()
```

This code can be found in: `cmpt140-ch04-py/cmpt140_ch04_function_calls_a/cmpt140_ch04_function_calls_a.pyde`.

- (b) Draw the moon with three craters anywhere on its surface

Solution

```
# CMPT 140 - Functions, Part One
# Topic: Function Calls
# Part (b)

size(500,500)

def draw_moon():
    """ draws moon of diameter 300px in centre of screen """
    ellipse(250,250,300,300)

def draw_crater(x,y):
    """
    draws crater of radius 30px with (x,y) centre coordinate
    x: x-coordinate of crater centre
    y: y-coordinate of crate centre
    """
    ellipse(x,y,30,30)

def draw_flag():
    """ draws flag planted at (205,295) """
    line(205,205,205,295)
    rect(205,205,50,30)

draw_moon()
draw_crater(200,200)
draw_crater(205,295)
draw_crater(325,275)
```

This code can be found in: `cmpt140-ch04-py/cmpt140_ch04_function_calls_b/cmpt140_ch04_function_calls_b.pyde`.

- (c) Draw the moon with a at least three craters; plant the flag in the centre of one of the craters and such that the flag is (partially) hiding another crater

Solution

```
# CMPT 140 - Functions, Part One
# Topic: Function Calls
# Part (c)

size(500,500)

def draw_moon():
    """ draws moon of diameter 300px in centre of screen """
    ellipse(250,250,300,300)

def draw_crater(x,y):
    """
    draws crater of radius 30px with (x,y) centre coordinate
    x: x-coordinate of crater centre
    y: y-coordinate of crate centre
    """
    ellipse(x,y,30,30)

def draw_flag():
    """ draws flag planted at (205,295) """
    line(205,205,205,295)
    rect(205,205,50,30)

draw_moon()
draw_crater(200,200)
draw_crater(205,295)
draw_crater(325,275)
draw_flag()
```

This code can be found in: `cmpt140-ch04-py/cmpt140_ch04_function_calls_c/cmpt140_ch04_function_calls_c.pyde`.

Exercise 2

What is the result of calling the following function?

```
def draw_something():
    ellipse(50,50,100,100)
    ellipse(25,35,20,20)
    ellipse(75,35,20,20)
    line(28,65,38,75)
    line(38,75,48,80)
    line(48,80,52,80)
    line(52,80,62,75)
    line(62,75,72,65)
```

Answer: The function `draw_something()` draws a smiling face to the Processing canvas. We can rename the function to something that reflects the fact that we are drawing a smiling face such as `draw_smiley_face()` since `draw_something()` can refer to drawing almost anything! We can insert comments prior to blocks of code which draw major, related portions of the face (e.g. eyes) to help break up the function and make it easier to understand and maintain.

Solution

```
# CMPT 140 - Functions, Part One
# Topic: Function Composition
```

```
def draw_smiley_face():

    # draw face
    ellipse(50,50,100,100)

    # draw eyes
    ellipse(25,35,20,20)
    ellipse(75,35,20,20)

    # draw smile
    line(28,65,38,75)
    line(38,75,48,80)
    line(48,80,52,80)
    line(52,80,62,75)
    line(62,75,72,65)

draw_smiley_face()
```

This code can be found in: `cmpt140-ch04-py/cmpt140_ch04_function_comp_lit/cmpt140_ch04_function_comp_lit.pyde`.

Exercise 3

Define Python functions for the following scenarios:

- (a) `draw_fixed_square()`: takes no parameters and draws a 50x50 square at coordinate (50,50)

Solution

```
# CMPT 140 - Functions, Part One
# Topic: Function Composition
# Part (a)
```

```
def draw_fixed_square():
    rect(50,50,50,50)
```

This code can be found in: `cmpt140-ch04-py/cmpt140_ch04_function_comp_a/cmpt140_ch04_function_comp_a.pyde`.

- (b) `draw_circle()`: takes parameters `x,y,d` and draws a circle centered at `(x,y)` with diameter `d`

Solution

```
# CMPT 140 - Functions, Part One
# Topic: Function Composition
# Part (b)
```

```
def draw_circle(x,y,d):
    ellipse(x,y,d,d)
```

```
draw_circle(20, 20, 20)
```

This code can be found in: `cmpt140-ch04-py/cmpt140_ch04_function_comp_b/cmpt140_ch04_function_comp_b.pyde`.

- (c) `draw_concentric_circles()`: takes parameters `x,y` and draws three circles on top of one another such that they all have centre coordinate `(x,y)`. The circles should have radius 90, 60, 30 in that order.

Solution

```
# CMPT 140 - Functions, Part One
# Topic: Function Composition
# Part (c)
```

```
def draw_circle(x,y,d):
    ellipse(x,y,d,d)
```

```
def draw_concentric_circles(x,y):
    draw_circle(x,y,180)
    draw_circle(x,y,120)
    draw_circle(x,y,60)
```

This code can be found in: `cmpt140-ch04-py/cmpt140_ch04_function_comp_c/cmpt140_ch04_function_comp_c.pyde`.

Exercise 4

Write documentation in the form of comments for the three functions you just wrote:

- (a) `draw_fixed_square()`

Solution

```
# CMPT 140 - Functions, Part One
# Topic: Function Composition
# Part (a)
```

```
def draw_fixed_square():
    """ draws a 50*50 square with top-left at (50,50) """
    rect(50,50,50,50)
```

This code can be found in: `cmpt140-ch04-py/cmpt140_ch04_function_doc_a/cmpt140_ch04_function_doc_a.pyde`.

- (b) `draw_circle(x, y, d)`

Solution

```
# CMPT 140 - Functions, Part One
# Topic: Function Composition
# Part (b)

def draw_circle(x,y,d):
    """
    draws a circle of diameter d centred at (x,y)
    x: circle's centre x-coordinate
    y: circle's centre y-coordinate
    d: circle's diameter in pixels
    """
    ellipse(x,y,d,d)
```

This code can be found in: [cmpt140-ch04-py/cmpt140_ch04_function_doc_b/cmpt140_ch04_function_doc_b.pyde](#).

(c) `draw_concentric_circles(x, y)`

Solution

```
# CMPT 140 - Functions, Part One
# Topic: Function Composition
# Part (c)

def draw_circle(x,y,d):
    """
    draws a circle of diameter d centred at (x,y)
    x: circle's centre x-coordinate
    y: circle's centre y-coordinate
    d: circle's diameter in pixels
    """
    ellipse(x,y,d,d)

def draw_concentric_circles(x,y):
    """
    draws three concentric circles of radius 90,60,30 at (x,y)
    x: circles' centre x-coordinate
    y: circles' centre y-coordinate
    """
    draw_circle(x,y,180)
    draw_circle(x,y,120)
    draw_circle(x,y,60)
```

This code can be found in: [cmpt140-ch04-py/cmpt140_ch04_function_doc_c/cmpt140_ch04_function_doc_c.pyde](#).