

# *FIT1008 – Intro to Computer Science*

## *Tutorial 7*

Semester 1, 2017

### *Objectives of this tutorial*

- To understand Classes and Objects.
- To understand namespaces and scoping.
- To understand memory diagrams.

### *Exercise 1*

This exercise and the next one are about scoping and namespaces.  
Examine these nested functions:

```
1 def silly():
2     x = 'ping'
3     def g():
4         print(x)
5     x = 'pong'
6     def f(x):
7         print(x)
8     g()
9     f(x)
10
11 silly()
```

We expected the silly() function call to print “ping” and then “pong”. What will it print? Why?

## Exercise 2

More scoping and namespaces. Now, with objects! A student named Juan is asked to make a Member class for a community pool. The only thing we care about is our pool's name, and our members' name, age and gender. So Juan writes:

```

1 class PoolMember:
2     poolname = "FIT1008_students_community_pool"
3     name = ""
4     age = 0
5     gender = None
6
7     def __init__(self, name, age, gender):
8         PoolMember.name = name
9         PoolMember.age = age
10        PoolMember.gender = gender

```

Juan makes himself the first member of the pool, and also its admin. It all seems to go so well!

```

1 >>> admin = PoolMember('Juan', 18, 'male')
2 >>> admin.name
3 'Juan'
4 >>> admin.age
5 18
6 >>> admin.gender
7 'male'
8 >>> admin.poolname
9 'FIT1008_students_community_pool'
10 >>>

```

But an order comes from above. The pool has to be open to any Monash students and staff, not just FIT1008 students. And we need to change the name before other students come! So we tell Juan to change the pool name before we make the new membership card for a well-respected professor, Emilia, who will be our supervisor:

```

1 >>> admin.poolname = "MONASH_all-inclusive_community_pool"
2 >>> admin.poolname
3 'MONASH_all-inclusive_community_pool'           # it looks alright
4 >>> supervisor = PoolMember(Emilia, 26, 'female') # let's make Emilia's user

```

Phew! Disaster averted! All seems alright again, until Emilia wants to check that the pool's name was really changed

```

1 >>> supervisor.poolname
2 'FIT1008_community_pool'           # what, this can't be!
3 >>> PoolMember.poolname           # let's check the global name
4 'FIT1008_community_pool'         # it's the same!

```

It's still the old name! Emilia wants to find who is responsible for the mistake, so she looks at the details for the admin:

```
1 >>> admin.name
2 'Emilia'
3 >>> admin.age
4 26
5 >> admin.gender
6 'female'
```

- What? Is Juan framing Emilia for his mistakes? Or is it just an unfortunate series of bugs?
- What did Juan do wrong? Can you write the correct class and the correct way to change the pool's name?

### Exercise 3

- Draw a memory diagram representing the state of memory after the following instructions have been executed.
- What is printed in the end?

```
1 numbers = [1, 2, 3 ]
2 other = numbers
3 numbers.append(4)
4 x = numbers[0]
5 x += 1
6 print(other)
7 print(numbers)
```