National University of Singapore
School of Computing
CS1010S: Programming Methodology
Semester I, 2016/2017

Tutorial 1
Introduction to CS1010S & Python

These questions will be discussed during the next week's discussion group meetings. Please be prepared to answer these questions during the session in class.

Note: Please check that your Python installation is configured correctly. See "Mission 0: Setting up Python" for more details.

1. Below is a sequence of expressions. What is the result printed by the interpreter in response to each expression? Assume that the sequence is to be evaluated in the order in which it is presented. You should determine the answers to this exercise without the help of a computer, and only later check your answers.

```
def square(x):
        return x ** 2

square

square(4)

square(square(square(2)))

f = square

f(2)

def try_f(f):
        return f(3)

try_f(f)

def f(z):
        return z

try_f(f)
```

   Observe that some of the examples shown above are indented and displayed over several lines. The indentation level of statements is significant in Python. As Python functions don't have explicity `begin` or `end`, and no curly braces to mark where the function code starts and stops. The only delimiter is a colon(:) and the indentation of the code itself.

2. First, using if-else, define a function `odd(x)` that returns *True* when its integer argument is an odd number and *False* otherwise.

   Now, without using if-else, define the function `new_odd(x)` that does the same.

3. Write a function that will return the number of digits in an integer.
4. Define a function that takes three numbers as arguments and returns the sum of the squares of the two larger numbers.
5. Write a function `is_leap_year` that takes one integer parameter and decides whether it corresponds to a leap year, i.e. the `is_leap_year` returns `True` if the input parameter is true, and `False` otherwise. So which years are leap years? Well, accordingly to Wikipedia:

   > In the Gregorian calendar, the current standard calendar in most of the world, most years that are integer multiples of 4 are leap years. In each leap year, the month of February has 29 days instead of 28. Adding an extra day to the calendar every four years compensates for the fact that a period of 365 days is shorter than a solar year by almost 6 hours. This calendar was first used in 1582.
   >
   > Some exceptions to this rule are required since the duration of a solar year is slightly less than 365.25 days. Over a period of four centuries, the accumulated error of adding a leap day every four years amounts to about three extra days. The Gregorian Calendar therefore omits 3 leap days every 400 years, omitting February 29 in the 3 century years (integer multiples of 100) that are not also integer multiples of 400.[3][4] For example, 1600 was a leap year, but 1700, 1800 and 1900 were not. Similarly, 2000 was a leap year, but 2100, 2200, and 2300 will not be. By this rule, the average number of days per year is 365 + 1/4 - 1/100 + 1/400 = 365.2425.