CSC148 Summer 2016 Quiz 01 (15 minutes)

First Name ………………………………… Last Name …………………………….. Lab room# ………………

In the back of this page see the API of class Rectangle; and, write answer to the following questions:

1. Create the docstrig for a method that verifies if this rectangle (self) fits in another rectangle. A rectangle fits in another rectangle if its width and height are smaller than those of the other rectangle. Implementing the method is NOT required.

```
def fits_in(self, other):
    """
    Return whether this rectangle fits in the other

    @param Rectangle self: this rectangle
    @param Rectangle other: another rectangle
    @rtype: bool

    >>> rec1 = Rectangle(100, 200, 400, 200)
    >>> rec2 = Rectangle(50, 100, 400, 200)
    >>> rec3 = Rectangle(50, 100, 500, 400)
    >>> rec1.fits_in(rec2)
    False
    >>> rec2.fits_in(rec1)
    False
    >>> rec1.fits_in(rec3)
    True

    """
    pass
```

2. Create a method that verifies if this rectangle (self) has the same area as another rectangle.

```
def has_same_area_as(self, other):
    """
    Return whether this rectangle has the same area as the other

    @param Rectangle self: this rectangle
    @param Rectangle other: another rectangle
    @rtype: bool

    >>> rec1 = Rectangle(100, 200, 400, 200)
    >>> rec2 = Rectangle(50, 100, 800, 100)
    >>> rec3 = Rectangle(50, 100, 500, 100)
    >>> rec1.has_same_area_as(rec2)
    True
    >>> rec1.has_same_area_as(rec3)
    False
    """
    return self.width * self.height == other.width * other.height
```

```python
class Rectangle:
    """"
    A rectangle defined by its top-left corner, width, and height

    Attributes:
    ===========
    @type x: int
        The x coordinate of rectangle's top-left corner
    @type y: int
        The y coordinate of rectangle's top-left corner
    @type width: int
        The width of rectangle
    @type height: int
        The height of rectangle

    """

    def __init__(self, the_x, the_y, the_width, the_height):
        """
        Create a new Rectangle self that with top_left corner at the_x and the_y
        and with width the_width and height the_height.

        the_x, the_y, the_width, and the_height are non-negative integers.

        @param Rectangle self: this rectangle
        @param int the_x: The x coordinate of rectangle's top-left corner
        @param int the_y: The y coordinate of rectangle's top-left corner
        @param int the_width: The width of rectangle
        @param int the_height: The height of rectangle
        @rtype: None
        """
        pass

    def __eq__(self, other):
        """
        Return whether this rectangle is equivalent to other.

        @param Rectangle self: this rectangle
        @param Rectangle | Any other: another rectangle
        @rtype: bool

        >>> rec1 = Rectangle(100, 200, 400, 200)
        >>> rec2 = Rectangle(50, 100, 400, 200)
        >>> rec3 = Rectangle(100, 200, 400, 200)
        >>> rec1 == rec2
        False
        >>> rec1 == rec3
        True
        """
        pass


    def translate_right(self, num):
        """
        Move Rectangle self to the right by a number of pixels.

        num is a non-negative integer.

        @param Rectangle self: this rectangle
        @param int num: the number of pixel the rectangle is translated
        @rtype: None

        >>> rec1 = Rectangle(100, 200, 300, 400)
        >>> rec1.translate_right(20)
        >>> rec1.x
        120
        """
        pass
```