ICA-3 Solutions

Problem 1

```
import math
class Point:

    def __init__(self, x, y):
        self._x = x
        self._y = y

    def translate(self, dx, dy):
        self._x += dx
        self._y += dy

    def distance_from_origin(self):
        return math.sqrt(self._x**2 + self._y**2)

    def as_list(self):
        return [self._x,self._y]

    def move_to(self, x, y):
        self._x = x
        self._y = y

    def __str__(self):
        return "(" + str(self._x) + "," + str(self._y) + ")"
```

Problem 2

```
class BookData:
    def __init__(self, author, title, rating):
        self._author = author
        self._title = title
        self._rating = rating

    def get_title(self):
        return self._title
    def get_author(self):
        return self._author
    def get_rating(self):
        return self._rating
    def __str__(self):
        return self._title + " : " + self._author + " : " +
str(self._rating)


def main():
    book_list = []
    prompt = ''
    while prompt != 'done':
        title = input("Book: " )
        author = input("Author: ")
        rating = int(input("Rating: "))
        b = BookData(author, title, rating)
        book_list.append(b)
        prompt = input()
    sum = 0
    for book in book_list:
        sum += book.get_rating()
        print(book)
    #Hey-we need an assert before using the print statement below!
    print("Average rating of all books: ", sum/len(book_list))

main()
```

Problem 3

```
class ClockTime:
    def __init__(self, hour, minutes, is_am):
        self._hour = int(hour)
        self._minutes = int(minutes)
        self._is_am = is_am

    def total_minutes(self):
        return self._hour * 60 + self._minutes

    def tick(self):
        if self._minutes < 59:
            self._minutes += 1
        else:
            """
            Increment the hour and handle the two cases where
            the hour is on the boundry of am/pm or pm/am
            """
            self._minutes = 0
            self._hour += 1
            if self._hour == 13:
                self._hour = 1
                self._is_am = not self._is_am

    def __str__(self):
        if self._is_am:
            am_pm = " AM"
        else:
            am_pm = " PM"
        mins = str(self._minutes)
        if self._minutes < 10:
            mins = "0" + mins
        return str(self._hour) + ":" + mins + am_pm
```