Aim: Write python programs to understand different object oriented features in python.

Define a class student with attributes roll number, name and score and methods to find grade. Grade is 'A' if score >=80, grade is 'B' if score >=60 and score<80, grade is 'C' if score >=50 and score<60 otherwise 'F'.

Objective of the Experiment:

1. Understanding classes and objects in python.

Source code for the implementation:

(Write only important functions)

Post Labs:

1. Consider the following code:

```
class Clock:
    def __init__(self, time):
        self.time = time
    def print_time(self):
        time = '6:30'print self.time
clock = Clock('5:30')clock.print_time()
```

(a) What does the code print out? If you aren't sure, create a Python file and run it.

(b) Is that what you expected? Why?

2. Consider the following code:

```
class Clock:
    def __init__(self, time):
        self.time = time
    def print_time(self, time):
        print time
clock = Clock('5:30')
clock.print_time('10:30')
```

(a) What does the code print out? If you aren't sure, create a Python file and run it.

(b) What does this tell you about giving parameters the same name as object attributes?

3. Consider the following code:

```
class Clock:
    def __init__(self, time):
        self.time = time
    def print_time(self):
        print self.time
boston_clock = Clock('5:30')
paris_clock = boston_clock
paris_clock.time = '10:30'
boston_clock.print_time()
```

(a) What does the code print out? If you aren't sure, create a Python file and run it.

(b) Why does it print what it does? (Are boston clock and paris clock different objects? Why or why not?)