(1번)

```
class Car:
    def __init__(self):
        self.color = 0xFF0000
        self.wheel_size = 16
        self.displacement = 2000

    def forward(self):
        pass

    def backward(self):
        pass

    def turn_left(self):
        pass

    def turn_right(self):
        pass

if __name__ == '__main__' :
    my_car = Car()

    print(type(my_car))

    print('0x{0:X}'.format(my_car.color))
    print(my_car.wheel_size)
    print(my_car.displacement)

    my_car.forward()
    my_car.backward()
    my_car.turn_left()
    my_car.turn_right()
```

(2번)

```
class ClassVar:
    class_text_list = []

    def __init__(self):
        self.instance_text_list=[]

    def add(self, text):
        self.class_text_list.append(text)
        self.instance_text_list.append(text)

    def print_list(self):
        print("class:",self.class_text_list)
        print("instance:",self.instance_text_list)

if __name__ == '__main__' :
    a = ClassVar()
    a.add('a')
    a.add('b')
    a.print_list()

    b = ClassVar()
    b.add('1')
    b.add('2')
    b.print_list()
```

(3번)

```python
class ContactInfo:
    def __init__(self, name, email):
        self.name = name
        self.email = email

    def print_info(self):
        print('{0}:{1}'.format(self.name, self.email))

if __name__ == "__main__" :
    inha = ContactInfo('인하', 'inha@inhatc.ac.kr')
    hanbit = ContactInfo('한빛', 'hanbit@inhatc.ac.kr')

    inha.print_info()
    hanbit.print_info()
```

(4번)

```
class Calculator :

    @staticmethod
    def plus(a,b) : return a + b

    @staticmethod
    def minus(a,b) : return a - b

    @staticmethod
    def multiply(a,b) : return a * b

    @staticmethod
    def divide(a,b) : return a / b

if __name__ == '__main__' :
    opr1 = 7
    opr2 = 2
    print("{0} + {1} = {2}".format(opr1, opr2, Calculator.plus(opr1,opr2)))
    print("{0} - {1} = {2}".format(opr1, opr2, Calculator.minus(opr1,opr2)))
    print("{0} * {1} = {2}".format(opr1, opr2, Calculator.multiply(opr1,opr2)))
    print("{0} / {1} = {2}".format(opr1, opr2, Calculator.divide(opr1,opr2)))
```

(5번)

```
class InstanceCounter:
    class_count = 0
    def __init__(self):
        InstanceCounter.class_count += 1
        self.instance_count = 1;

    @classmethod
    def print_class_count(cls) :
        print(cls.class_count)

    @staticmethod
    def print_static_count():
        print(InstanceCounter.class_count)

    #instancemethod
    def print_instance_count(self):
        print(self.instance_count)

if __name__ == '__main__' :

    a = InstanceCounter()
    a.print_class_count()
    a.print_static_count()
    a.print_instance_count();
    InstanceCounter.print_class_count()
    InstanceCounter.print_static_count()
    #InstanceCounter.print_instance_count()

    b = InstanceCounter()
    b.print_class_count()
    b.print_static_count()
    b.print_instance_count();
    InstanceCounter.print_class_count()
    InstanceCounter.print_static_count()
    #InstanceCounter.print_instance_count()
```

(6번)

```
class HasPrivate :
    def __init__(self):
        self.public = "PUBLIC"
        self.__private = "PRIVATE"

    def print_internal(self):
        print(self.public)
        print(self.__private)

obj = HasPrivate()
obj.print_internal()
print(obj.public)
print(obj.__private)
```