```python
# ============== capsule  ==============
balance = 8000

def deposit(money):
    global balance
    balance += money

def inquire():
    print("잔액은 %d원입니다." % balance)

deposit(1000)
inquire()

# ============== account  ==============
class Account:
    def __init__(self, balance):
        self.balance = balance
    def deposit(self, money):
        self.balance += money
    def inquire(self):
        print("잔액은 %d원입니다." % self.balance)

sinhan = Account(8000)
sinhan.deposit(1000)
sinhan.inquire()

nonghyup = Account(1200000)
nonghyup.inquire()

# ============== human  ==============
class Human:
    def __init__(self, age, name):
        self.age = age
        self.name = name
    def intro(self):
        print(str(self.age) + "살 " + self.name + "입니다.")

kim = Human(29, "홍길동")
kim.intro()
lee = Human(45, "이승우")
lee.intro()
```

```python
# =============== student  ===============
class Human:
    def __init__(self, age, name):
        self.age = age
        self.name = name

    def intro(self):
        print(str(self.age) + "살 " + self.name + "입니다")


class Student(Human):
    def __init__(self, age, name, stunum):
        super().__init__(age, name)
        self.stunum = stunum

    def intro(self):
        super().intro()
        print("학번 : " + str(self.stunum))

    def study(self):
        print("하늘천 따지 검을현 누를황")


kim = Human(29, "홍길동")
kim.intro()
lee = Student(34, "이승우", 930011)
lee.intro()
lee.study()

# =============== getset  ===============
class Date:
    def __init__(self, month):
        self.month = month
    def getmonth(self):
        return self.month
    def setmonth(self, month):
        if 1 <= month <= 12:
            self.month = month


today = Date(8)
```

```python
today.setmonth(15)
print(today.getmonth())

# =============== property  ===============
class Date:
    def __init__(self, month):
        self.inner_month = month
    def getmonth(self):
        return self.inner_month
    def setmonth(self, month):
        if 1<= month <= 12:
            self.inner_month = month
    month = property(getmonth, setmonth)

today = Date(8)
today.month = 15
print(today.month)

# =============== property2  ===============
class Date:
    def __init__(self, month):
        self.inner_month = month
    @property
    def month(self):
        return self.inner_month
    @month.setter
    def month(self, month):
        if 1 <= month <= 12:
            self.inner_month = month

today = Date(8)
today.month = 15
print(today.month)

# =============== hidden  ===============
class Date:
    def __init__(self, month):
        self.__month = month
    def getmonth(self):
        return self.__month
    def setmonth(self, month):
```

```python
        if 1 <= month <= 12:
            self.__month = month
    month = property(getmonth, setmonth)

today = Date(8)
today.__month = 15
print(today.month)

# =============== classmethod  ===============
class Car:
    count = 0
    def __init__(self, name):
        self.name = name
        Car.count += 1
    @classmethod
    def outcount(cls):
        print(cls.count)

pride = Car("프라이드")
korando = Car("코란도")
Car.outcount()

# =============== staticmethod  ===============
class Car:
    @staticmethod
    def hello():
        print("오늘도 안전 운행 합시다.")
    count = 0
    def __init__(self, name):
        self.name = name
        Car.count += 1
    @classmethod
    def outcount(cls):
        print(cls.count)

Car.hello()

# =============== eqop  ===============
class Human:
    def __init__(self, age, name):
        self.age = age
```

```python
        self.name = name
    def __eq__(self, other):
        return self.age == other.age and self.name == other.name

kim = Human(29, "홍길동")
sang = Human(29, "홍길동")
moon = Human(44, "문종민")
print(kim == sang)
print(kim == moon)

# ============== clsstr  ==============
class Human:
    def __init__(self, age, name):
        self.age = age
        self.name = name
    def __str__(self):
        return "이름 %s, 나이 %d" % (self.name, self.age)

kim = Human(29, "홍길동")
print(kim)

# ============== clslen  ==============
class Human:
    def __init__(self, age, name):
        self.age = age
        self.name = name
    def __len__(self):
        return self.age

kim = Human(29, "홍길동")
print(len(kim))
```