```
from abc import ABC , abstractmethod
class Vichicle(ABC):
  @abstractmethod
  def start(self):
    print('the car starts')
  @abstractmethod
  def stop(self):
    print('the car stops')

class BMW(Vichicle):
  def start(self):
    print('BMW starts!')
  def stop(Self):
    print('BMW stop!')
car = BMW()
car.start()
car.stop()
```

```
BMW starts!
BMW stop!
```

```
# make a class of payment system
class PaymentSystem(ABC):
  @abstractmethod
  def pay(self):
    print('the payment completed successfully!')

# credit , debit , upi , paytm

class CreditCard(PaymentSystem):
  def pay(self):
    print("the payment has been done by the credit card")

class DebitCard(PaymentSystem):
  def pay(self):
    print("the payment has been done by the debit card")

class Paytm(PaymentSystem):
  def pay(self):
    print("the payment has been done by the paytm")

debit = DebitCard()
debit.pay()
credit = CreditCard()
credit.pay()
paytm = Paytm()
paytm.pay()
```

```
the payment has been done by the debit card
the payment has been done by the credit card
the payment has been done by the paytm
```

```
# create an abstract method class TransportSrvice
# book_ticket
# cancel_ticket
from abc import ABC , abstractmethod

class TransportService(ABC):
  @abstractmethod
  def book_ticket(self):
    print("ticket are booked!")
  @abstractmethod
  def cancel_ticket(self):
    print("ticket's are cancel!")


class Flight(TransportService):
  def book_ticket(self):
    print("the ticket has been booked")

  def cancel_ticket(self):
    print("the ticket has been canceled!")

class Bus(TransportService):
  def book_ticket(self):
    print("the ticket has been booked")

  def cancel_ticket(self):
    print("the ticket has been canceled!")
```

```
class Train(TransportService):
  def book_ticket(self):
    print("the ticket has been booked")

  def cancel_ticket(self):
    print("the ticket has been canceled!")

flight = Flight()
flight.book_ticket()
flight.cancel_ticket()

bus = Bus()
bus.book_ticket()
bus.cancel_ticket()

train = Train()
train.book_ticket()
train.cancel_ticket()
```

```
the ticket has been booked
the ticket has been canceled!
the ticket has been booked
the ticket has been canceled!
the ticket has been booked
the ticket has been canceled!
```

```
# Modules -- its are the .py files which contains some methods,class and variables
# packages -- its are the collection of the modules and that also contain __init__.py file

# folder --
          # |
          # |
          # userdata.py
          # admindata.py
          # paymentdata.py
          # __init__.py

# Types of package in python
# -- regular packages
# -- namespace packages


# function types in packages
# -- Built - in function
#      -- os : interfaces for intranction with the operating system
#      -- math : mathematical function
#      -- datetime : basic date and time types
#      -- JSON : json encoder and decoder
#      -- tkinter : the standard python interface to the TCL / TK GUI tool kit

# -- User - defined function
#       -- NUMPY : for numerical computing and multi-dimensional arrays.
#       -- PANDAS : for data manipulation and analysis using dataframes and series.
#       -- SCIKIT-LEARN : for classical machine learning algorithms like classification and regression
#       -- TENSORFLOW & PYTourch: for deep learning and neural networks
```

Start coding or generate with AI.