Problem 1:Design a course registration platform
Objects and Behaviors:

Computer:
    Data:
    Behaviors:startUp,connectToInternet

Internet:
    Data:Collection of Websites
    Behaviors: searchForElectricityWebsites

Electricity Website:
    Data:URL,ListOfClasses
    Behaviors:search,display,sort,compare

User:
    Data: Name,Phone,Address,emailAddress
    Behaviors:startUpComputer,search,register,cancel,review

Sequence of invoking behaviors on objects
Registering class on class registration platform

```
Class registration platform;
User kai;
kai.startUpComputer->pin:computerStartUp
If the Internet. isAvailable
    kai.loginToThePlatform->userName,pin:login
    kai.searchForClass->classDescription:listOfClass
    kai.findDesirableClass->className:classDetail
    If class.hasOpenSeats
        kai.findDesirableSession->classSession:classTimeAndLocation
        If session.hasOpenSeats
            kai.register->personalInformation:confirmation
        else
            searchOtherSession
        end
    else
        searchOtherClass
end
```

Problem 2:Design a food delivery app

Objects and Behaviors:

Internet Service:

Data: Name,phoneNumber

Behaviors: connect

App:

Data: ListOfRestaurant,listOfDriver,map

Behaviors:

User:

Data: Name,Phone,userLocation

Behaviors: logInToApp,orderFood,reviews,cancel,contactDriver

Food Driver:

Data: Name,driverPhoto,driveLicense,Phone,driverLocation

Behaviors: pickUpFood,contactUser

Restaurant:

Data: food menu, price

Behaviors: confirmUserOrder, prepareFood, contactDrive

Traffic System:

Data: RoadCondition, MilesOfRoad

Behaviors: getRoadCondition,getDeliveyTime

Map System:

Data: road

Behaviors: locateUser,locateDriver,askTrafficSystem,selectBestRoad

Credit Card:

Data: Name, bankName, cardNumber, address, securityCode, expiry

Behaviors:

Bank:

Data:

Behaviors: authorizedTranscation

Help Service:

Data: servicePhoneNumber

Behaviors: chatOnLine, Question, Answer

Sequence of invoking behaviors on objects

orderingFoodInApp

```
User kai;
restaurant;
food driver;
App Uber Eat;

if the Internet. isAvailable
kai.loginToApp->userName, pin: connected
if ordering food now
    kai.orderFood->location:nearbyRestaurant
    if nearbyRestaurant. is Available
        kai.findDesirebleRestaurabt->listOfRestaurant:Restaurant
        kai. findDesirebleFood->listOfFood:Food
        kai. placeTheOrder->creditCard,address:confirmation
        trafficSystem.getgetDeliveryTime->roadName: predictedDeliveryTime
        Loop
        if hourOfPredictedDeliveryTime<0.5
            return predictedDeliveryTimeIsGood
        else
            mapSystem.changeRoad
        end

    end
else
    kai.setLocationToOrderFood->location:listOfRestaurabt
    kai.findDesirableRestaurant->listOfRestaurant: Restaurant
    kai.findDesirableFood->listOfFood:Food
    kai.placeTheOrder->creditCard,address:confirmation
end
bank.authorizedTransction
```

Problem 3 Design a platform for buying tickets of local events
Objects and behaviors:

```
Computer:
    Data:
    Behaviors: startUp, connectToInternet

Internet:
    Data:Collection of Websites
    Behaviors:searchForElectricityWebsites

Electricity Website:
    Data:URL,listOfEvent
    Behaviors:search,display,sort,compare

Map System:
    Data:Road
    Behaviors:locateUser

User:
    Data:Name,Phone,Address
    Behaviors:startUpComputer,search,filter,purchase,cancel,review
```

Sequence of invoking behaviors on objects
BuyingTicketsOfLocalEvents

```
Buyer kai;
kai.startUpComputer->pin:computerStartUp
If the Internet. isAvailable
    kai.loginToThePlatform->userName,pin:login
    kai.searchForLocalEvents->location:localEvents
    kai.findDesirableEvents->eventName:EventDetail
    If event.hasVacantTicket
        kai.placeTheOrder->creditCard,address:confirmation
    else
        SearchOtherEvents
    end
end
bank.authorizeTransaction
```

problem 4 Buy a computer from Amazon

Objects and behaviors:

Computer:
    Data:
    Behaviors: startUp, connectToInternet

Internet:
    Data: Amazon
    Behaviors: searchForElectricityWebsites

Electricity Website:
    Data:URL, ListOfComputerStore
    Behaviors:search,display,sort,compare

Computer Buyer:
    Data: Name, Phone, Address, AmazonAccount
    Behaviors:startUpComputer,search,filter,review,compare, cancel

ComputerStore:
    Data: Name, listOfComputer, ComputerDetails: brand, size, price,quality
    Behaviors: login, post, confirmOrder, delivery

Credit Card:
    Data: Name, bankName, cardNumber, address, securityCode, expiry
    Behaviors:

Bank:
    Data:
    Behavior: authorizedTransaction

Help Service:
    Data: servicePhoneNumber
    Behaviors: chatOnLine, Question, Answer

Sequence of invoking behaviors on objects

Buying computer on Amazon

```
Website Amazon;
Buyer kai;

kai. startUpComputer->pin: computerStartUp
if the Internet. isAvailable
    kai.loginToAmazon-> userName, pin:login
    kai.searchForComputer-> ComputerDescription: listOfComputers
    if Amazon. hasComputer
        kai.findDesirableComputer->ComputerDescription:price,quality
        kai.purchase->creditCars,address:confirmation
    else
        Not purchase computer on Amazon
    end
end
bank.authorizeTransaction
```

problem 5: Design an app for booking hotels.

objects and behaviors:

```
Internet Service:
    Data:Name
    Behaviors:connect

App:
    Data:listOfHotels,map
    Behaviors:

User:
    Data:Name,Phone,numberOfPeople,numberOfRooms,destination,data
    Behaviors:logInToApp,serchHotel,reviews,cancel,contactHotelOwner

Hotel Owner:
    Data:Name,Phone
    Behaviors:confirmUserOrder,contactUser

Hotel System:
```

Data: room
Behaviors: getRoomCondition

Credit Card:
Data:Name,bankName,cardNumber,securityNumber,address,expiry
Behavior:

Bank:
Data:
Behavior:authorizedTransnction

Help Service:
Data: servicePhoneNumber
Behavior:chatOnLine,Question,Answer

Sequence of invoking behaviors on objects
bookingHotelInApp

```
User kai;
App bookinghotel;



if the Internet. isAvailable
kai.loginToApp ->userName,pin:connected
if booking hotel now
    kai.bookHotel->destination, date: nearbyHotel
        if nearbyHotel.isAvailable
            kai.findDesireableHotel->listOfHotel:Hotel
            if NumbersOfRooms>0
                return RoomIsAvaiable
                kai.findDesireableTypeOfRooms->listOfRoom:Room
            if  NumberOfThisTypeOfRoom>0
                return ThisTypeOfRoomIsAvaiable
                kai.placeTheOrder->creditCard,address:confirmation
            else
                App.askKaiChangeTypeOfRoom
            end
```

```
else
        Kai.setDestinationAndDateToBookHotel->destination,date:ListOfHotel
        Kai.findDesirableHotel->listOfHotel:Hotel
        kai.findDesirableTypeOfRoom->listOfRoom:Room
        kai.placeTheOrder->creditCard,address:confirmation
end
bank.authorizeTransaction
```