Problem 1: Design a course registration platform

Objects and behaviors:

Computer:

- Data:
- Behaviors: startUp, connectToInternet

Internet:

- Data:
- Behaviors: searchWebsite

Website:

- Data: courseRegistration, listOfCourse
- Behaviors: Search

Student:

- Data: Name, Phone, Address, EmailAddress, Schedule
- Behaviors: startUpComputer, addCourse, dropCourse, compareCourse, Submit, Cancel, Confirm

School:

- Data:
- Behaviors: publishCourse, login, cancel

Course:

- Data: Status, Faculty, CRN, Name, Time,
- Behaviors: Publish, Add, Drop

Sequence of invoking behaviors on objects

PostingCourseInThePlatform

```
School NU;
if the Internet.isAvailable
    NU.loginToThePlaform -> userName, pin : login
    if the Website.isAvailable
        NU.publishCourse -> Course
    else
        refresh the Website
    end
end
```

RegistingCourseInThePlatform

```
Student new;
Course List;
new.startUpComputer -> pin : computerStartUp
if the Internet.isAvailable
    new.loginToThePlaform -> userName, pin : login
    new.searchForCourse -> courseDescription : listOfCourse
    if CourseStatus.isAvailable && StudentSchedule.isAvailable
        new.findDesirableCourse - > courseName : Course
        new.addCourse -> CRN : Course
        if student need drop Course
            new.dropCourse -> CRN : Course
        end
    else
        return errorReport
    end
else
    return errorReport
end
```

Problem 2: Order food in a food delivery app

Objects and behaviors:

Internet Service:
Data: Name, phoneNumber
Behaviors: connect

App:
Data: listOfCar, listOfDriver, map
Behaviors: confirmOrder

User:
Data: Name, Phone, userLocation, Destination, numberOfPassager, tripTime
Behaviors: logInToApp, callDelivery, reviews, cancel, contactDriver

Driver:
Data: Name, driverPhoto, driverLicence, Phone, driverLocation
Behaviors: confirmUserOrder, contactUser, startDelivery

Traffic System:
Data: roadCondition
Behaviors: getRoadCondition

Vehicles:
Data: numberOfVehicles
Behaviors: Boolean isMetWithAccident()

Map System:
Data: Road
Behaviors: locateUser, locateDriver, askTrafficSystem, selectBestRoad

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

Bank:
Data:
Behaviors: authorizedTransaction

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

Sequence of invoking behaviors on objects

```
callingDeliveryInApp

User new;
Driver driver;
App uber;

if the Internet. isAvailable
    new.loginToApp -> userName, pin: connected

if calling Delivery now
    new.callDelivery -> location, destination: nearbyDelivery
    if nearbyDelivery.isAvailable
        new.findDesirebleCar -> listOfCar : Car
        new.placeTheOrder -> creditCard, address : confirmation
        trafficSystem.getRoadCondition -> roadName : roadCondition
        Loop
        if numberOfVehicles < setNumberWhenConditionIsGood
            return roadConditionIsGood
        else
            mapSystem.changeRoad
        end
    else
        App.askNewChangeLocation
    end
else
    new.setTimeToUseCar -> date, time : listOfCar
    new.findDesirableCar -> listOfCar : Car
    new.placeTheOrder -> creditCard, address : confirmation
end

uber.confirmOrder
if Order is confirmed
    driver.startDelivery
end

Bank bank;
bank.authorizeTransaction
```

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

Objects and behaviors:

Computer:

Data:

Behaviors: startUp, connectToInternet

Internet:

Data: TicketSource, Eventbrite, TicketTailor

Behaviors: searchForElectricityWebsites

Website:

Data: URL, ListOfCompany

Behaviors: search, display, sort, compare

Buyer:

Data: Name, Phone, Address, EmailAddress

Behaviors: startUpComputer, search, filter, apply, reviews, cancel, sort, compare

TicketCompany:

Data: Name, Phone, eventDueDate, Price

Behaviors: login, post, confirmApplication

Sequence of invoking behaviors on objects

```
PostingEventInThePlatform

TicketCompany TicketSource;
if the Internet. isAvailable
        TicketSource.loginToTheplatform -> userName, pin : connected
        if theWebsite. isAvailable
                TicketSource.post -> EventDetails : confirmationForPosting
        else
                refresh the Website
        end
end

TicketCompany Eventbrite;
Redo PostingEventInThePlatform

TicketCompany TicketTailor;
Redo PostingEventInThePlatform


SearchingAndOrderEventInThePlatform

Buyer new;
new.compare -> Price : TicketCompany

Switch (new.chooseChannel)
case: TicketSource
TicketCompany TicketSource;
new. startUpComputer -> pin : computerStartUp
if the Internet. isAvailable
        new.loginToThePlatform -> userName, pin : login
        new.searchForTicketCompany -> TicketCompanyDescription : ListOfCompanies
        new.findDesirableTicketCompany -> TicketCompanyName : TicketCompanyWebsite
        TicketSource = website
        new.searchForEventInTicketSource -> EventDescription : ListOfEvents
        if TicketSource. hasVacantPositions
                new.findDesirableEvent -> EventName : EventWebsite
                new.apply -> personalData  : confirmation
        else
                searchOtherCompanies //jump to other company
        end
end

case: Eventbrite
repeat
case: TicketTailor
repeat
```

Problem 4: Buy a computer from Amazon

Objects and behaviors:

Computer:
  Data:
  Behaviors: startUp, connectToInternet

Internet:
  Data: Microsoft, Apple, Dell
  Behaviors: searchForElectricityWebsites

Website:
  Data: URL, ListOfCompany
  Behaviors: search, display, sort, compare

Buyer:
  Data: Name, Phone, Address, EmailAddress
  Behaviors: startUpComputer, search, filter, apply, reviews, cancel, sort, compare

ComputerCompany:
  Data: Name, Phone, eventDueDate, Price
  Behaviors: login, post, confirmApplication

Sequence of invoking behaviors on objects

PostingComputerInThePlatform

```
ComputerCompany Microsoft;
if the Internet. isAvailable
    Microsoft.loginToTheplatform -> userName, pin : connected
    if theWebsite. isAvailable
        Microsoft.post -> computerDetails : confirmationForPosting
    else
        refresh the Website
    end
end

ComputerCompany Apple;
Redo PostingComputerInThePlatform

ComputerCompany Dell;
Redo PostingComputerInThePlatform
```

SearchingAndOrderComputerInThePlatform

```
Buyer new;
new.compare -> Price : ComputerCompany

Switch (new.chooseBrand)
case: Microsoft
ComputerCompany Microsoft;
new. startUpComputer -> pin : computerStartUp
if the Internet. isAvailable
    new.loginToThePlatform -> userName, pin : login
    new. searchForComputerCompany -> ComputerCompanyDescription : ListOfCompanies
    new.findDesirableComputerCompany -> ComputerCompanyName : ComputerCompanyWebsite
    Microsoft = website
    new.searchForComputerInMicrosoft -> computerDescription : ListOfComputers
    if Microsoft. hasSuitableComputer
        new.findDesirableComputer -> ComputerName : ComputerWebsite
        new.apply -> personalData  : confirmation
    else
        searchOtherCompanies //jump to other company
    end
end

case: Apple
repeat
case: Dell
repeat
```

Problem 5: Design an app for booking hotels

Objects and behaviors:

Computer:

Data:

Behaviors: startUp, connectToInternet

Internet:

Data: GrandHyatt, FourSeasons, Hilton
: Collection of Websites

Behaviors: searchForElectricityWebsites

Website:

Data: URL, ListOfCompany

Behaviors: search, display, sort, compare

Buyer:

Data: Name, Phone, Address, EmailAddress, personID

Behaviors: startUpComputer, search, filter, apply, reviews, cancel, sort, compare

HotelCompany:

Data: Name, Phone, eventDueDate, Price

Behaviors: login, post, confirmApplication

Sequence of invoking behaviors on objects

PostingEventInThePlatform

```
HotelCompany GrandHyatt;
if the Internet. isAvailable
        GrandHyatt.loginToThePlatform -> userName, pin : connected
        if theWebsite. isAvailable
                GrandHyatt.post -> BookingDetails : confirmationForPosting
        else
                refresh the Website
        end
end

HotelCompany FourSeasons;
Redo PostingBookingInThePlatform

HotelCompany Hilton;
Redo PostingBookingInThePlatform
```

SearchingAndOrderBookingInThePlatform

```
Buyer new;
new.compare -> Price : HotelCompany

Switch (new.chooseCompany)
case: GrandHyatt
HotelCompany GrandHyatt;
new. startUpComputer -> pin : computerStartUp
if the Internet. isAvailable
        new.loginToThePlatform -> userName, pin : login
        new. searchForHotelCompany -> HotelCompanyDescription : ListOfCompanies
        new.findDesirableHotelCompany -> HotelCompanyName : HotelCompanyWebsite
        GrandHyatt = website
        new.searchForBookingInGrandHyatt -> BookingDescription : ListOfBookings
        if GrandHyatt. hasVacantPositions
                new.findDesirableBooking -> BookingName : BookingWebsite
                new.apply -> allPersonalData  : confirmation
        else
                searchOtherCompanies //jump to other company
        end
end

case: FourSeasons
repeat
case: Hilton
repeat
```