Question 1 Design a course registration platform

Class, object and behaviors:

Class Internet
  Object web
    Information:
    Behaviors: isConnect(university)

Class University
  Object neu
    Information: universityName, currentStudentsNames, availableCourseSelectTime, studentsRequiredCredits
    Behaviors: isCurrentNEUStudent(student), isCourseSelectTime(), isCreditsEnough(student)

Class Student
  Object larry
    Information: name, university, major, credits, schedule
    Behaviors: select(course), isScheduleAvailable(schedule), addSchedule(course.schedule), addCredits(course)

Class Course
  Object java
    Information: courseSchedule, courseCredits, majorRequire, passerNames, fee, courseStudentsNumber
    Behaviors: isRightMajor(student), noPassedBefore(student), isFull()

Class Bank
  Object account
    Information:
    Behaviors: add(course.fee)

Sequence of invoking behaviors on objects

```
Internet web = new Internet;
University neu = new University;
Student larry = new Student;
Course java = new Course;
Bank account = new Bank;

if web.isConnect(neu) -> universityName
   if neu.isCurrentNEUStudent(larry) -> currentStudentsNames, name
      if neu.isCourseSelectTime() -> availableCourseSelectTime
         while not neu.isCreditsEnough(larry) -> studentsRequiredCredits, credits
            larry.select(course) -> java
               if not java.isFull() -> courseStudentsNumber
                  if java.isRightMajor(larry) -> majorRequire, major
                     if java.noPassedBefore(larry) -> passerNames, name
                        if larry.isScheduleAvailable(schedule) -> schedule, courseSchedule
                           larry.addSchedule(java.courseSchedule) -> schedule, courseSchedule
                           addCredits(java.courseCredits) -> credits, courseCredits
                           account.add(java.fee) -> fee
                        else
                           continue
                     else
                       continue
                  else
                     continue
               else
                  continue
      else
         print “Not a good time.”
   else
      print “You are not current student.”
else
   print “Unstable Internet.”
```

Question 2 Order food in a food delivery app

Class, object and behaviors:

Class Internet
- Object web
  - Information:
  - Behaviors: connect

Class Application
- Object app
  - Information:
  - Behaviors: show, calBookFee, showMapDetail, showReceipt, provideAssess, sendOrder

Class Customer
- Object larry
  - Information: name, adress, account
  - Behaviors: register, order, openApp

Class Deliveryman
- Object kobe
  - Information: deliName, deliAccount
  - Behaviors: orderReceiving, clickFinish

Class Restaurant
- Object mcdonald
  - Information: menu, restaurantAdress, coursePrice, restaurantsNames
  - Behaviors: provide, cook

Class Bank
- Object bank
  - Information:
  - Behaviors: deduct, add

Sequence of invoking behaviors on objects

Internet web = new Internet
Application app = new Application
Customer larry = new Customer
Deliveryman kobe = new Deliveryman
Restaurant mcdonald = new Restaurant
Bank bank = new Bank

Web.connect
mcdonald.provide -> restaurantsNames, menu, restaurantAdress, coursePrice
app.show -> restaurantsNames, menu, restaurantAdress, coursePrice
larry.openApp
larry.register -> name, adress
larry.order -> menu
    app.calBookFee -> adress, restaurantAdress, coursePrice
    app.sendOrder
    mcdonald.cook
    kobe.orderReceiving -> restaurantAdress, adress
    if kobe.clickFinish
        app.showReceipt
        bank.deduct(larry.account)
        bank.add(kobe.deliAccount)
        app.provideAssess
    else
        app.showMapDetail

Question 3 Design a platform for buying tickets of local events

Class, object and behaviors:

Class Internet
  Object web
    Information:
    Behaviors: connect

Class App
  Object app
    Information:
    Behaviors: show, receiveOrder, provideOrder, sendPayRequest, showTicket, limitPayTime

Class Event
  Object event
    Information: ticketPrice, eventTime
    Behaviors: provideEvent, sendTicket, splitTicket

Class LocalNews
  Object news
    Information: sortedEventInfo
    Behaviors: sortInformation, provideInformation

Class Bank
  Object bank
    Information:
    Behaviors: sendReceipt

Sequence of invoking behaviors on objects

Internet web = new Internet
App app = new App
Event event = new Event
LocalNews news = new LocalNews
Bank bank = new Bank

Web.connect
event.provideEvent -> ticketPrice, eventTime
news.sortInformation -> ticketPrice, eventTime
news.provideInformation -> sortedEventInfo
app.show -> sortedEventInfo
if app.receiveOrder
    app.sendPreOrder
    event.splitTicket
    app.sendPayRequest
    if bank.sendReceipt
        app.provideOrder
    else
        app.limitPayTime
        continue
    event.sendTicket
    app.showTicket

Question 4 Buy a computer from Amazon

Class, object and behaviors:

Class Internet
  Object web
    Information:
    Behaviors: connect

Class Customer
  Object larry
    Information: amazonAccount, pin, address
    Behaviors: search, order, login, clickReceive

Class Ecommerce
  Object amazon
    Information:
    Behaviors: show, calBookFee, sendPayRequest, sendOrder, showOrderDetail, giveAssess

Class Seller
  Object apple
    Information: computerType, price, appleAddress
    Behaviors: giveInformation, ship

Class Bank
  Object bank
    Information:
    Behaviors: sendReceipt

Sequence of invoking behaviors on objects

Internet web = new Internet
Customer larry = new Customer
Ecommerce amazon = new Ecommerce
Seller apple = new Seller
Bank bank = new Bank

web.connect
apple.giveInformation -> computerType, price, appleAddress
amazon.show -> computerType, price
larry.login -> amazonAccount, pin, address
larry.search -> computerType, price
larry.order
    amazon.sentPayRequest
    if bank.sendReceipt
        amazon.sendOrder
        apple.ship
    while not larry.clickReceive
        amazon.showOrderDetail
    amazon.giveAssess

Question 5 Design an app for booking hotels

Class, object and behaviors:

Class Internet
  Object web
    Information:
    Behaviors: connect

Class App
  Object app
    Information:
    Behaviors: show, sendPayRequest, sendPreOrder, limitPayTime

Class Customer
  Object larry
    Information: appAccount, pin, address, price, type
    Behaviors: login, search, order

Class Hotel
  Object hotel
    Information: roomPrice, hotelAddress, roomType
    Behaviors: giveInformation, splitRoom, giveRoom

Class Bank
  Object bank
    Information:
    Behaviors: giveReceipt

Sequence of invoking behaviors on objects

Internet web = new Internet
App app = new App
Customer larry = new Customer
Hotel hotel = new Hotel
Bank bank = new Bank

web.connect
hotel.giveInformation -> roomPrice, hotelAddress, roomType
app.show -> roomPrice, address, roomType
larry.login -> appAccount, pin
larry.search -> address, price, type
larry.order
    app.sendPayRequest
    app.sendPreOrder
    hotel.splitRoom
    if bank.giveReceipt
        app.showReceipt
        hotel.giveRoom
    else
        app.limitPayTime