Introduction

Several student have commented that they found some difficulties understanding how data structures are set in memory. These four exercises are given as additional practice.

Students can solve them using pen and paper. The goal is to read the code, understand it, and produce diagrams that explain the state of variables (simple and complex types) in memory at every numbered point (in comments). The points are not numbered in any particular order, and the program may go through each point more than once.

If the students find them confusing, or they want to confirm that their understanding is right, they are encourage to show their proposed solutions to the lecturers to receive feedback on the first three exercises. The fourth exercise is a marked assignment (see below).

Graded assignment

The three first programs are provided as additional practice for students. The fourth one is the first graded assignment for this course. Students should submit their solutions on paper. Pen-drawing is fine as long as **it is clear**; there is no need for fancy computer-generated diagrams. To submit, hand your diagrams to any member of the teaching body: Peng Du, Sergio Gutierrez-Santos, or Sokratis Karkalas. Do not forget to put your name in all the pages.

Deadlines

Full-time students Thursday 25th October 2012, 21h59 (London time).

Part-time students Thursday 22th November 2012, 21h59 (London time).

1 Car dealing

```
class Car{
  String Model;
  double price;
}

void trade(Car newCar, Car oldCar, int myFund){
  if(newCar.price <= oldCar.price+myFund){
        println "we have a deal"
  } else {
        println "forget about it..."
  }
  println "  now we try to be naughty"
  Car temp = new Car();
  temp = newCar;
  newCar = oldCar;
  oldCar = temp;
  // Point 1
}

Car myOldFord = new Car();
myOldFord.Model = "Ka";
myOldFord.price = 2000;

Car fancyRacer = new Car();
fancyRacer.Model = "911";
fancyRacer.price = 300000;

int myBudget = 10000;
// Point 2
```

```
println "Let’s exchange cars! Deal?"
trade(fancyRacer,myOldFord,myBudget);
// Point 3
println "The new car has become " + fancyRacer.model;
println "And the old car has become " + myOldFord.model;
println "The deal has failed";
```

2 Film rating

```
class Film{
    String title;
    int reviewer = 0;
    double totalStar = 0;

    void review(int stars){
       reviewer++;
       totalStar += stars;
    }

    double getRating(){
       if (reviewer==0) {
           return 0;
       } else {
           return totalStar / reviewer;
       }
    }
}

void rate(Film f, int stars){
    println "Viewer voting being processed...";
    f.review(stars);
    f = null;
    stars++;
    println "The value of star inside this method is: " + stars;
    // Point 1
}

Film helpMeChoose(Film f1, Film f2) {
    if (f1.rating > f2.rating) {
       return f1;
    } else {
       return f2;
    }
}

Film film1 = new Film();
film1.title = "Prometheus";
Film film2 = new Film();
film2.title = "Twilight";
print film1.reviewer + " reviewers have given " + film1.title
println  " an average rating of " + film1.getRating() + " stars.";
print film2.reviewer + " reviewers have given " + film2.title
println  " an average rating of " + film2.getRating() + " stars.";
// Point 2
int stars = 5;
rate(film1,stars);
```

```
println "The value of star here is now: " + stars;
stars = 4;
rate(film1,stars);
println "The value of star here is now: " + stars;
stars = 2;
rate(film2,stars);
println "The value of star here is now: " + stars;
stars = 3;
rate(film2,stars);
println "The value of star here is now: " + stars;
// Point 3
print film1.reviewer + " reviewers have given " + film1.title
println " an average rating of " + film1.rating + " stars.";
print film2.reviewer + " reviewers have given " + film2.title
println " an average rating of " + film2.rating + " stars.";
Film finalChoice = helpMeChoose(film1,film2);
println "And we officially recommend: " + finalChoice.title;
// Point 4
```

3 People, people. . .

```
class Person {
  String firstname;
  String surname;

  Person father;
  Person mother;

  String personDetails() {
        String result="";
        result+="firstname: "+firstname;
        result+="\n";
        result+="surname: "+surname;
        return result;
  }

  String familyDetails() {
        String result="";
        result+="father\n";
        result+="------\n";
        result+=father.personDetails();
        result+="\n";
        result+="mother\n";
        result+="------\n";
        result+=mother.personDetails();
        return result;
  }

  void showDetails() {
        println personDetails();
        println familyDetails();
  }
}

Person p1=new Person()
p1.firstname="Adam"
```

```
p1.surname="Taylor"
// Point 1
Person p2=new Person()
p2.firstname="Eve"
p2.surname="Smith"
// Point 2
Person p3=new Person()
p3.firstname="Daniel"
p3.surname="Taylor"
p3.father=p1
p3.mother=p2
p3.showDetails()
// Point 3
```

4 Public transport

```
class Travelcard {
    String type;
    int zone;
}

class OysterCard {
    boolean autotopup;
    boolean registered;
    double amount;
    Travelcard travelcardAttached;
}

boolean yellowReader(OysterCard card, int zone) {
    double fare = getFare(zone);
    double cost = 0.0;

    if(card.travelcardAttached != null) {
       if(card.travelcardAttached.zone >= zone) {
          return true;
       }
    }

    if (card.amount > fare) {
        card.amount -= fare;
        return true;
    }

    return false;
}

double getFare(int zone) {
    double fare=0.0;

    if (zone <= 2) {
      fare = 2.3;
    } else if (zone <= 4) {
      fare = 4.6;
    } else {
      fare = 6.2;
    }
    return fare
}

OysterCard card1 = new OysterCard();
card1.autotopup = true;
card1.registered = false;
card1.amount = 50.0;
card1.travelcardAttached = null;
// Point 1
Travelcard tcard = new Travelcard();
tcard.type = "week";
tcard.zone = 2;
// Point 2
OysterCard card2 = new OysterCard();
card2.autotopup = false;
card2.registered = true;
card2.amount = 6.3;
card2.travelcardAttached = tcard;
// Point 3
println yellowReader(card1, 2)? "pass" : "problem with card";
println yellowReader(card1, 6)? "pass" : "problem with card";
println yellowReader(card2, 4)? "pass" : "problem with card";
println yellowReader(card2, 5)? "pass" : "problem with card";
```