There is no quiz or handout today. Please just review the

following important topics that were already presented in

the lecture.

------------------------------------------------------------------

1. Please review the ways to copy structures with external

members. Please see the file copies.pdf, which is a scan of

correct memory diagrams for this example.

(In those pictures, the thing on the left is the "call stack",

which is where local variables go. The thing on the right is

the "heap" which is where objects go.)

Here is the example:

class Vegetable {

public String name;

public int calories;

}

For each of the following show them the code and then draw

pictures of the memory diagram (stack and heap) for each of the

three types of copying. Again, see the file copies.pdf for

corect memory diagrams.

Assume that v is a vegetable object, and we want to make a copy.

/\* 1. Reference copy \*/

Vegetable x = v;

/\* 2. Shallow copy \*/

Vegetable x = new Vegetable();

x.name = v.name;

x.calories = v.calories;

/\* 3. Deep copy \*/

Vegetable x = new Vegetable();

x.name = new String(v.name);

x.calories = v.calories;

----------------------------------------------------------------

Review: Which copy is best if we want x and v to be independent of

one another?

Reference copy is not good because either member could be changed

with one variable and that change would be reflected in the other.

Shallow copy is PERFECT

Deep copy is not necessary in this example because Strings are

immutable. (Why waste time/space with a deep copy when there is

no advantage.)

----------------------------------------------------------------

Please review 2-dimensional ragged arrays of references using some

example that you like.

One idea would be:

public class Calendar {

private String[][] appointments;

public Calendar() {

appointments = new String[7][]; // 7 days of the week

for (int i = 0; i < 7; i++) {

appointments[i] = new String[0]; // empty array

}

}

public void addAppointment(String appointment, int day) {

String[] temp = new String[appointments[day].length + 1];

for (int i = 0; i < appointments[day].length; i++) {

temp[i] = appointments[day][i];

}

temp[temp.length - 1] = appointment;

appointments[day] = temp;

}

public void display() {

for (int i = 0; i < appointments.length; i++) {

System.out.println("DAY " + (i + 1) + ":");

for (int j = 0; j < appointments[i].length; j++) {

System.out.println(appointments[i][j]);

}

System.out.println();

}

}

public static void main(String[] args) {

Calendar c = new Calendar();

c.addAppointment("Get groceries", 2);

c.addAppointment("Bake cake", 5);

c.addAppointment("Defeat enemy", 2);

c.addAppointment("Pick apples", 6);

c.display();

}

}

-----------------------------------------------------------------------

If any time remains, please ask the students if they have questions

or if there are some other topics that they would like for you to

review.