**Chapter 10**

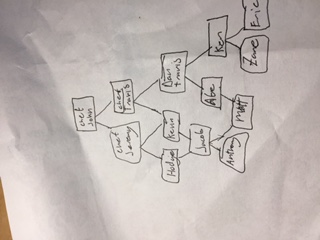
Ex. 10.1 – Open networ-v1m Create a newsfeed object and enter some posts into feed and then display the feed’s comments.

Done

Ex 10.2 – Create a MessagePost object and enter it into the newsfeed. Add a comment to the messagePost object. When you list the news feed again it will the post have a comment attached? Explain the behavior you observe.

The post in the news feed will have no comments to begin with. When we use the show method it calls another method in the display which is part of the MessagePost class. It is this method that will print out the name of the author, their message but it doesn’t print the comments associated with it.

Ex 10.3 – Draw an inheritance hierarchy of somewhere you know.



Ex. 10.4 – Open the network-v2 project. What are some changes you can see in the class diagram? Remove the “extend Post” phrase and close the editor. What changes do you see now? Add the phrase back in.

Before making any edits there is a different type of arrow between the Post, MessagePost, PhotoPost. This type of arrow shows there is an inheritance relation between the classes. Once extend Post is remove the arrow disappears and the class stands alone as it no longer inherits from the other. When we put it back in the diagram goes to how it was when we opened the project.

Ex. 10.5 – Can you call inherited methods? What do you observe about the inherited methods?

Yes, you can call inherited methods. When we right click on the instance we see drop down tabs that hold all inherited methods from a given class.

Ex. 10.6 – Add a printShortSummary method to both photo and message Post. It should print the message/photo and author.

Done and saved, but wouldn’t it make more sense to have this be completely contained in Post as the code is currently identical in each class?

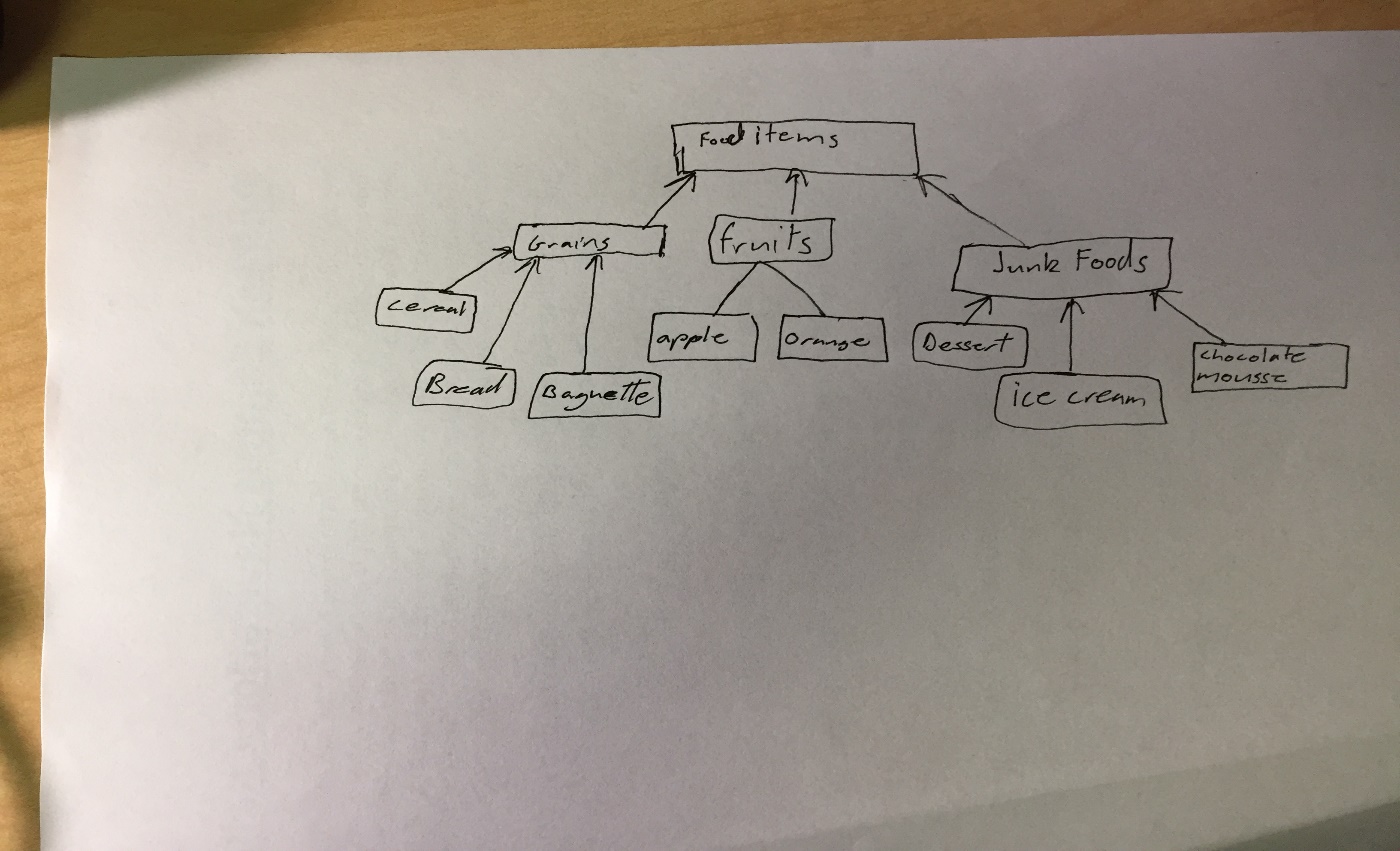
Ex. 10.7 – Set a break point in the first line of the MessagePost class’s constructor. Step through using the step into to get through the code. Observe the instance fields and their initialization. Describe what you see.

The first thing passed when stepping through the constructor is super(author);. When I step into it calls the super classes constructor and initializes the fields message, username, timestamp, likes, and comments. After this it, moves to initializing local fields of the class.

Ex. 10.8 – Add a class for event posts to the project.

Done and saved.

Ex. 10.9 – Order the items into an inheritance hierarchy: apple, ice cream, bread, fruit, food item, cereal, orange, dessert, chocolate mousse, baguette.

Ex. 10.10 – In what inheritance hierarchy might a touch pad and a mouse be?

It could be in a UserInputObject or perhaps a ScreenNavigator are possible super classes for a mouse and touch pad.

Ex. 10.11 – In what kind of inheritance relationship are rectangle and square?

A rectangle would be a super class to a square because by definition a square will always be a rectangle as but a rectangle won’t always be a square.

Ex. 10.12 – A) Which of the following assignments are legal why and why not?

Person p1 = new Student(): - legal because student inherits class person.

Person p2 = new PhDStudent(); - legal because it is a subclass of student and student is a subclass of person.

PhDStudent phd1 = new Student(); - Not legal because student is a super class of PhdStudent.

Teacher t1 = new Person();- Not legal teacher is not a super class of Person.

Student s1 = new PhDStudent();- Legal, PhDStudent is a subclass of Student.

B) Suppose that we the following legal declarations and assignments.

Person p1 = new Person();

Person p2 = new Person();

PhDStudent phd1 = new PhDStudent();

Teacher t1 = new Teacher();

Student s1 = new Student();

Based on those just mentioned which of the following assignments are legal? Why or why not?

s1 = p1 - Not legal, Student is a subclass of Person

s1 = p2 – See previous answer.

p1 = s1 – Legal, because Student is a subclass of Person

t1 = s1 – Not legal, because Teacher isn’t a subclass of Student.

s1 = phd1 – Legal, PhDStudent is a subclass of Student.

phd1= s1 – Not legal, because PhDStudent isn’t a superclass to student.

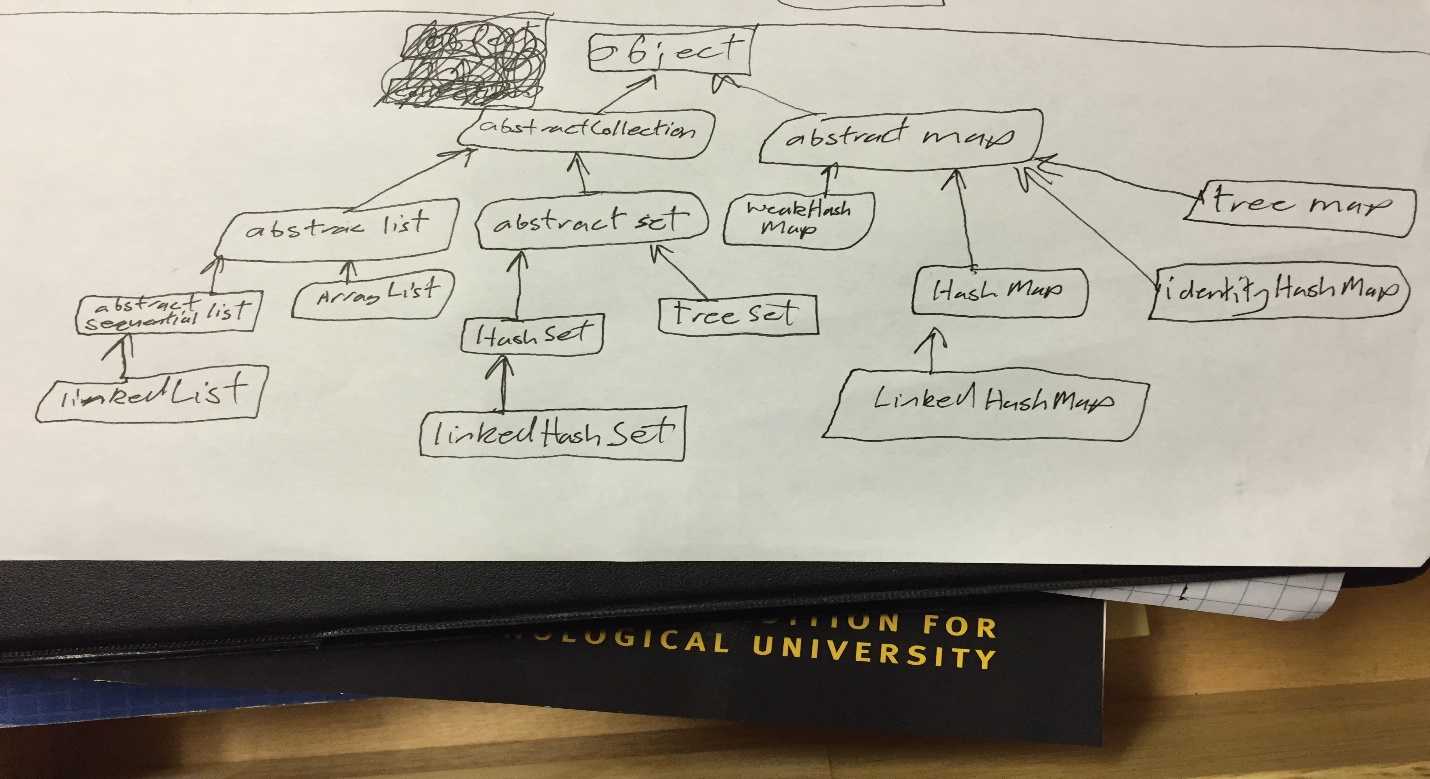
Ex. 10.13– Create bare bones version of these classes in blueJ and test the combinations.

Done and saved, view class TestStuff code for the tests.

Ex. 10.14– What has to change in the NewsFeed class when another subclass is added to the Post subclass?

Thanks to the inheritance nothing should need to be changed in the NewsFeed class because it treats everything as a post. Since all xxxpost classes are subclasses of Post they can be treated as if they were post objects.

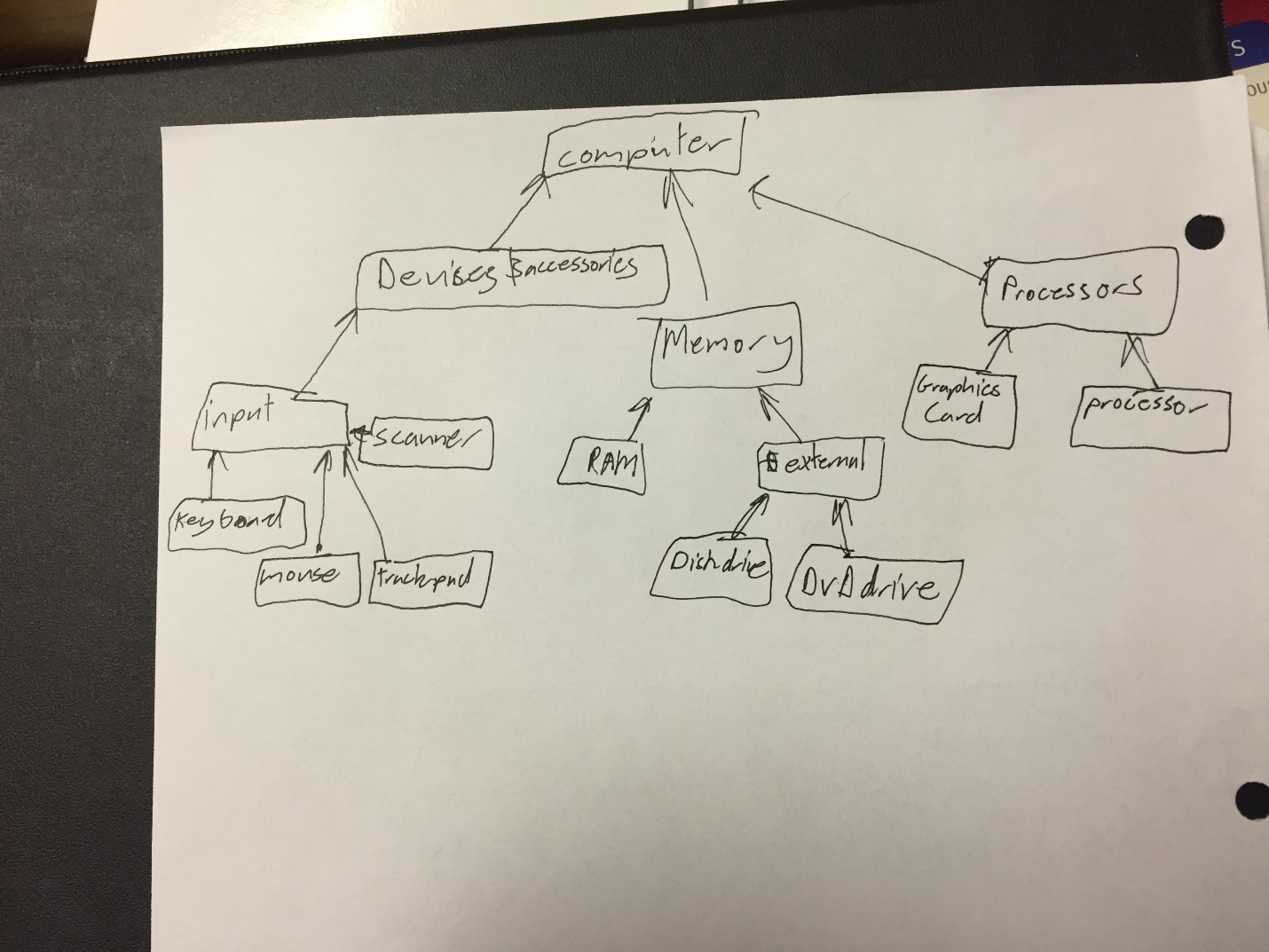
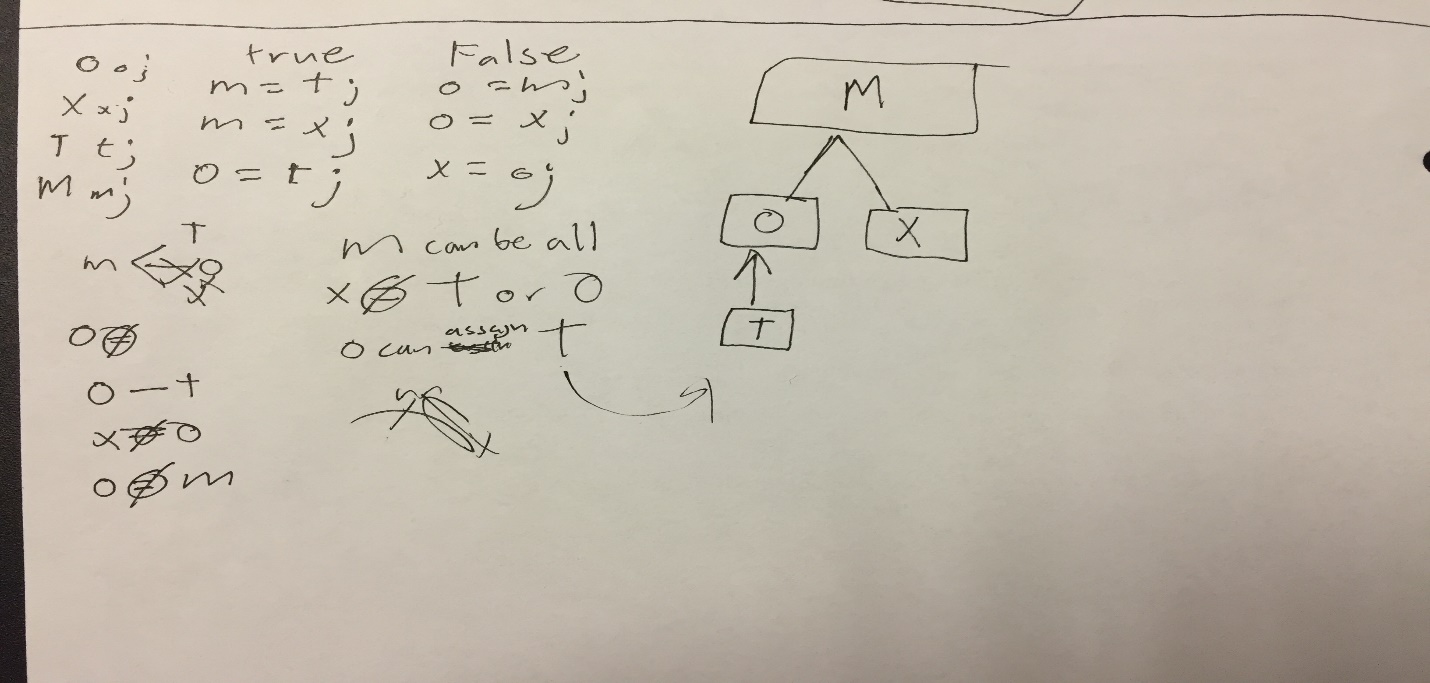
Ex. 10.15– Use the Documentation of the Java standard class libraries to find out about the inheritance hierarchy of the collection classes. Draw a diagram showing the hierarchy.



Ex. 10.16 – In lab-classes project from chapter 1 add an instructor to the project. Use inheritance to avoid code duplication. There can only be one instructor per lab class.

Done and Saved.

Ex. 10.17 – Draw and inheritance hierarchy representing the parts of a computer system. (processor, memory, disk drive, DVD drive, printer, scanner, keyboard, mouse, etc. )

Ex. 10.18 – From the code in the text, Draw a class diagram detailing their relationships

Ex. 10.19– Draw and inheritance hierarchy of abstract list and all its subclasses as they are defined in the Java standard library.

