Exam 1 Study Guide

ArrayLists –

Be able to use built-in ArrayList methods to identify, locate and access a given element.

Inheritance –

You'll be asked to write code to develop an inheritance hierarchy (super class / sub class.)

For example:

Given the Savings sub-class that follows, create the super-class named 'Account':

- Make the Account class abstract.
- Data variables in the Account class are 'name' (type String which can be accessed directly by the sub-class but not by the outside world,) and 'balance' (type double which cannot be accessed by any sub-class.)
- Add the methods that are referenced in the Savings sub-class.
- Make it so that the 'deductFees' method is known to the Account class, but the implementation for deductFees is required to be in the Savings class.

```
public class Savings extends Account
{
    public Savings(String name, double balance)
    {
        super(name, balance);
    }
    public void deductFees()
    {
        setBalance(getBalance() * .50);
    }
    public String toString()
    {
        //  Show the account type, depositor's name and balance...
        return "Savings account for " + super.toString();
    }
}
```

```
//  Create your Account class here…
```

Be familiar with the rules regarding inheritance (T/F questions – see quiz 1.)

Interfaces –

Be able to apply a user-written interface to a concrete implementation.

Exceptions –

True / False questions about the rules of exception handling, exception-related Java reserved words, inheritances in exceptions, etc.

There may be some tracing of methods through exception handling, showing output.