What did we learn in Lab 03?

- What kinds of memory problems can valgrind report?
- Why is proper memory management important?
- Is there any leak in the following code fragment? What is wrong with this code (memory-wise)?

```cpp
int *a[10];
int x = 4;
a[1] = &x;
cout<< *a[1]<<endl;
delete a;
```

- What is a garbage collector? (We did not learn this one! I just want to encourage you to look it up!)
- What is the advantage of using references instead of pointers in C++ (from a memory perspective)?
What we are going to learn in Lab 04?

• What is inheritance useful for?
• What is the difference between public, private, and protected members of a class from an inheritance standpoint?
• What are parameterized types and functions? What are they useful for?
• What kinds of optimization can the compiler do with templates?
• What is dynamic binding?
• What is static binding?
• What is a pure virtual function?
• You have three different hierarchy of classes (like the figure below). Someone asks you to make a list of instances of these classes. How would you do that?