

Lab Exercises. (Due: beginning of your lab during the week of 30 September 2002) This week's lab is concerned with Project #1. The intent is to make sure you are on the right path, and to encourage you *not* to put things off to the last instant. I suggest, therefore, that you aim to get as much of this lab as possible completed by the 24 September labs.

Copy the directory `$master/hw/lab4` to a directory of your own, using commands such as

```
cp -r $master/hw/lab4 .
```

In this directory, you should find a file called `README`, with directions about what to do during the scheduled lab. We strongly suggest that you read over this file *before* going to your lab.

We intend that you finish the lab exercises *in lab* and have your TA check them off.

Homework Exercises. (Due: Fri., 27 September 2002 at midnight) Create a directory to hold your answers to this homework set. Copy the files from `$master/hw/hw4` into this directory. Use the command `submit hw4` to submit your solutions to the problems below.

1. Implement the following *without* using the operators `+`, `-`, `/`, `*`, or `%`, `++`, `--`, `+=`, `-=`, `*=`, `/=`, `%=`, or any other classes from the Java class library or elsewhere. Just use the bitwise operators `&`, `|`, `^` and `~`, `<<`, `>>`, `>>>`, and `&=`, etc. You may include one `for` loop with constant bounds (such as `for (i = 0; i < 32; i += 1)...`). The usual admonition applies: *Don't fight the problem!*

```
public class Adder {
    /** The value x+y. */
    public static int add(int x, int y) { /* FILL IN */ }
}
```

2. Assuming `x` and `y` are both `ints`, what does the following do and why?

```
x ^= y; // NOTE: Same as x = x ^ y
y ^= x;
x ^= y;
```

Demonstrate it empirically by filling in the template `xor.java` in the `hw4` directory.

3. I want to create a class called `Unique` such that there is exactly one object of type `Unique`—one is always created (with `new`), and no more can be created after that. Furthermore, I want any attempt at creating additional `Uniques` to be caught by the compiler (i.e., before the program is ever run). Devise a way to do this in Java and give an example. This is a thought problem, so just put your answer in an ordinary text file called `hw4.txt`.

4. A singly linked list structure (like `IntList`) can be circular. That is, some element in the list can have a tail (next) field that points to an item *earlier* in the list (not necessarily to the first element in the list). Come up with a way to detect whether there is such a circularity somewhere in a list. Do *not*, however, perform any **new** operations, or otherwise create new objects, and don't modify any existing objects in the process. Use just simple list pointers without changing any fields of any list. See `List.java` in the `hw4` directory.

5. The `java.util.Iterator` interface is intended to provide a general way to sequence through a collection of things without knowing anything about the data structure in which it is stored. A class (let's say `Stuff`) that wants to provide a way to iterate through the collection of things that it contains (let's say they are `Items`) typically provides a method called, appropriately enough, `iterator`, which returns some kind of `Iterator`. Clients can then write loops like this:

```
Stuff c = something that creates a Stuff;
Iterator p = c.iterator ();
while (p.hasNext ()) {
    Item i = (Item) p.next ();
    do something to i;
}
```

The file `Stuff.java` in the `hw4` directory contains a partial example of such a class. Fill in its `iterator` method, and add any other classes you need, so that the client main program in `Stuffer.java` works properly. The `Iterator` interface is in the on-line Java standard library documentation, and also on page 24 of the *Data Structures (Into Java)* reader.