

CS 61b - Summer 2005

Homework #1 - Due June 27 at the beginning of class

This homework is to be done individually. You may, of course, ask your fellow classmates for help if you have trouble editing files, compiling programs, or using the turnin program. Do not share code or specific parts of answers. Please see the syllabus for guidelines on appropriate interaction with your fellow classmates.

Copy the homework 1 files to your directory by typing the following from your account:

```
mkdir hw1
cd hw1
cp $master/hw/hw1/* .
```

Homeworks will often require written answers as well as small programs (Java source files). *All written answers are to be placed in a file called **writeup.txt***. This file should be a plain text file. Make sure this file is readable by “wordpad” or “emacs” or similar text-only editor. No credit will be given for written answers stored in a non-text format. Please ask your TA for help if you have trouble.

When you are finished with the homework, you will submit it electronically. First put all of the files you want to submit into their own directory (including writeup.txt). Change to that directory, and then type “submit hw1”

If you can't complete the entire homework, and it complains about you not having certain files, then just edit a blank file with that name, and then save it. This will submit an empty file, but the submit program should then work.

Example:

```
cory [12] ~/tmp > submit hw1
Submitting writeup.txt.
Submitting SubtotalAdderA.java.
Submitting SubtotalAdderB.java.
Submitting FibonacciPrinter.java.
Looking for other files to turn in....
Turn in ./foo? [yes/no] no
Turn in ./bar? [yes/no] no
The files you have submitted are:
    ./FibonacciPrinter.java ./SubtotalAdderA.java
./SubtotalAdderB.java ./writeup.txt
Is this correct? [yes/no] y
Submitting to cs61b-ra ....
Submission complete.
```

Problem 1 (on or after 21-Jun) (13 points)

Weiss, 1.11, 1.12, 1.14 (put your answers to these in writeup.txt)

There are usually several reasonable solutions to a given problem. You are going to design a “Subtotal Addition Calculator” using two different algorithms.

A subtotal addition calculator is the modern version of the old-timey paper roll addition machines that accountants used to use. The user enters a list of integers, followed by the “subtotal” key. Once the subtotal key is pressed, the calculator shows the sum of the numbers entered so far. The user can then enter another “subtotal group”, followed by the subtotal key, and the calculator will print out the sum of that group. This process continues until the user hits the “total” key and the sum of all numbers entered is displayed.

In our program, we will use the number “0” to represent both the subtotal key and the total key.

Here is an example session with the subtotal group calculator (user input is in **bold**):

```
Subtotal Addition tool

Enter one number per line; enter '0' to indicate the end of
a subtotal group
Enter two lines consisting of '0' to terminate
3
8
2
0
Subtotal: 13
6
7
0
Subtotal: 13
1
1
9
7
2
0
Subtotal: 20
0
Total: 46
```

Let's consider one way of decomposing this problem:

```
loop1:  
While there are more subtotal groups {  
    process a group;  
    add its subtotal to the running total;  
}
```

```
loop2 (how to process a group):  
set subtotal to 0;  
until we get a zero {  
    add the number we get to the running subtotal;  
}
```

In this case, we have a loop within a loop.
Let's consider another decomposition:

```
repeat {  
    read a number;  
    do the right thing, updating subtotal or total as appropriate;  
}
```

In this case we have only one loop. But within that loop, we do a variety of things (either print out the subtotal, print out the total, or exit the program).

You are to write Java code that implements the two pseudocode decompositions above. The files `SubtotalAdderA.java` and `SubtotalAdderB.java` are the starting points. Find the comment section in each file and place your code there. There is already pseudocode that you can use as a starting point.

The two decompositions should have the same behavior for any given set of input. Don't worry about handling error cases or bad user input. We'll talk about how to do that next week.

Question: Which decomposition seems most intuitive? The loop within a loop, or the single loop? Why? (put the answer to this question into `writeup.txt`)

Submit `SubtotalAdderA.java` and `SubtotalAdderB.java`

Problem 3 (on or after 22-Jun) (10 points)

Use the program in Figure 2.6 of Weiss and the file `FibonacciPrinter.java` as starting points for a Fibonacci number printer. Your program will perform the following:

- User types in a number N followed by carriage-return
- Your program prints out the Nth Fibonacci number
- This continues until the user enters a blank line by just typing carriage-return, at which point your program terminates

Recall that you can compute the Nth Fibonacci number using the following recursive functions:

$$F(1) = F(2) = 1$$

$$F(N) = F(N-1) + F(N-2)$$

First few Fibonacci numbers: {1, 1, 2, 3, 5, 8, 13, 21, 34, ...}

Example of the user interface:

```
$ java FibonacciPrinter
Enter any number of numbers, one per line;
Terminate with empty line:
3
2
17
1597
5
5
```

Done reading

```
$
```

Your program must satisfy the following constraint: **It must only compute a particular Fibonacci number once.** Your program must satisfy this constraint by computing the Fibonacci numbers “on demand”, and storing the results into a (dynamically resized) array. Thus, if a user asks for the 15th Fibonacci number, you can compute the first 15 Fibonacci numbers and store them into an array. If they then ask for the 7th number, simply return the pre-computed value from the array. If your array is too small, resize it in a manner similar to Weiss Figure 2.6

It is ok if your program crashes on bad input (We’ll cover how to gracefully handle bad input later in the semester). You may find the following code snippet useful for your design:

```
int numRequested = (int) Integer.parseInt(oneLine)
```

Think carefully about the difference between array sizes and array subscript values. What value should you pass to the `resize()` method: N, or N-1? Submit `FibonacciPrinter.java`.

Problem 4 (on or after 23-Jun) (4 points)

These answers go into writeup.txt

4.1 Weiss 3.1

4.2 Why does Java use packages to organize classes, when there is an implicit organization in terms of the class hierarchy (via subclassing?)

4.3 What is the difference between these two class definitions?

```
class Foo { ... }  
public class Foo { ... }
```

When would you want to use one over the other?