Lab 2

Due 22 February

Handout 2
CSCI 136: Spring, 2004
16 February

The Silver Dollar Game

1 Warm-up Exercises

There is nothing to hand in for this part, but make sure you understand these examples.

1. What is printed by the following program?

```
class Example {
    public int num;

    public Example(int initial) {
        num = initial;
    }

    public int getNum() {
        num = num + 1;
        return num;
    }

    public static void main(String args[]) {
        Example first = new Example(10);
        Example second = new Example(20);
        System.out.println(first.getNum());
        System.out.println(second.getNum());
    }
}
```

What would be printed if we changed the declaration of `num` to be

```
public static int num;
```

What does this tell you about `static` fields?

2. Browse the javadoc documentation available from the class web page. Look up the `structure` package's `Vector` class. Become familiar with how to look up information on these webpages.

2 Short Answers

Answer the following questions from the book (SC refers to the Self Check Problems):

SC 3.2
SC 3.3
SC 3.4
3.1
3.6 (don't write the class— just answer what the advantage would be)

Hand these in at the beginning of class on Wednesday (day of lab).

3 Lab Program

This week, we will write the Silver Dollar Game at the end of Chapter 3. You should come to lab with a written design of the program.

The internal representation of the coin strip should be a Vector, but you will need to decide what sort of information should be stored in the Vector. Make sure your representation supports all of the operations necessary, ie. testing for a legal move, printing the board, testing for a win, moving pieces easily, etc. You should think about alternative designs and be able to justify your decisions.

Once you have decided on a representation, write down the set of operations supported by your data structure. In other words, what are the public methods of `CoinStrip`, and what are their pre-conditions and post-conditions?

The document you bring to lab should include both a description of the representation and the operations on it. This initial design will constitute a fraction of your grade on the lab assignment.

The `main` method of this class should provide create a `CoinStrip` and then prompt each of two players to make their moves. A move will be of the form:

```
cn ns
```

where `cn` is the coin to be moved and `ns` is the number of spaces it should be moved. The program should prompt the user for another input if the move is illegal.

To read input from a window, please use the class `ReadStream` from the structures library. The method `readInt` will read the next integer typed in the window, skipping all white space that may come first. See the structures library documentation for more information.

When you are finished with the program, answer the thought questions at the end of the lab. Put your answers in a comment at the top of your file, which you should name `CoinStrip.java`. The file should be contained in a folder Lab2*yourName*. Turn in the folder by dragging it into the dropoff folder on Cortland.

As in all labs, you will be graded on design, documentation, style, and correctness. Be sure to document your program with appropriate comments, including a general description at the top of the file, a description of each method with pre- and post-conditions where appropriate. Also use comments and descriptive variable names to clarify sections of the code which may not be clear to someone trying to understand it.

This program is due by 11:59pm on Sunday, 22 February.