COLBY COLLEGE
Department of Mathematics
and Computer Science

CS115 **Allen B. Downey**
Spring 1998

Assignment 3: Conditional execution

Due: Wednesday 25 February

Reading: Kamin, Mickunas and Reingold, Chapter 4 (except 4.5) and 8.1

1 Programming assignment

The purpose of this assignment is to practice using conditional execution constructs and class (static) methods.

Print ordinal numbers

1. Create a new project named Ordinal.π. Put it in a new folder named Ordinal, and name the class Ordinal. Make it an Application (not an Applet).
2. Use a nested if structure to print ordinal values as on the quiz. Note that the implementation I showed on the quiz was not quite correct, because it would not handle numbers in the teens correctly.

HINT: You will need to find the rightmost digit of a number and the rightmost two digits. You can use the modulus operator (%) to do that. For example, what's 2467457845 % 10?

Print the date

1. At the beginning of main, add the following two lines:

```
int day = 13;
int month = 7;
```

2. Use the code from the previous section to print "The 13th day of the 7th month." Of course, you have to write it so that it will work for any day of the month and any month.

FOOD FOR THOUGHT: To do this part, you probably had to make two copies of a really complicated if statement. I hope you had it completely debugged, because now if you have to make any changes, you will have to make them in two places.

What if we wanted to print "The 2nd minute of the 13th hour of the 7th day of the 11th month of the 1998th year?" What would your code look like then?

Make your code into a method

A much more sensible way to proceed is to make your code into a method. That way you can have one copy of the code and invoke it several times with different arguments each time.

1. After the declaration of `main`, create a new method definition. It will look a lot like the definition of `main`, except that the name will be `printOrdinal` and the arguments will be different. It should look like:

   ```
   public static void printOrdinal (int x) {
        // put your code here
   }
   ```

2. For now, you can ignore the words `public` and `static`, but you should know that `void` means that this method does not return anything, that `printOrdinal` is the name of the method, and that `(int x)` means that this method takes a single argument named `x` that has type `int`.

   IMPORTANT: Don't forget the word `static`. If you leave it out, you will get a mystifying error message. As a matter of fact, you should go ahead and do it on purpose, just so you see what the error message is. Then if you do it by accident, you'll know what it means.

3. Move your code from `main` into the body of the new method. You will have to make a couple of changes. For example, you should not set the value of `x` in `printOrdinal` — you will receive it as an argument from main.

4. Modify `main` to invoke `printOrdinal` twice, printing "The 13th day of the 7th month" just like it did before.

Other than being shorter, what advantages does this program have over the previous version?

Make your program even nicer

1. In `printOrdinal`, replace the nested if with a switch statement. Isn't that better?

One more method!

1. Read Section 8.1 of the text.
2. Create a new project named Fraction.π, in the usual way. Make it an Application.
3. Write the method `fractionOfDay` described in Exercise 4, at the end of Section 8.1.
4. Write a `main` method that tests your method by invoking it with various arguments and printing the results.

What to turn in

1. Clean up your code as always, correcting the indentation and adding comments. In future assignments, I will not remind you to do this, but I will expect it.
2. Print copies of the final version of `printOrdinal` and `fractionOfDay`, and write your name in the upper right-hand corner.