

CS100M Spring 2006 Project 5 due Thursday 4/13 at 6pm

Turn off the file backup feature in DrJava—it causes problems on some system configurations! Go to menu item **Edit**→**Preferences**, choose the last category, “Miscellaneous,” then *uncheck* the box for “Keep Emacs-style Backup Files.”

Submit your files `api.txt`, `Complex.java`, `Fractal.java`, and `C Calculator.txt` on-line in CMS under Project 5 before the project deadline. For java code be careful to submit the `.java` file, not the `.class` file. Both correctness and good programming style contribute to your project score.

You must work either on your own or with one partner. You may discuss background issues and general solution strategies with others, but the project you submit must be the work of just you (and your partner). If you work with a partner, you and your partner must register as a group in CMS and submit your work as a group.

Objectives

In this project, you will learn to implement classes and use objects, and you will learn about the Java API (Application Programming Interface) by looking through parts of the on-line Java API documentation. In Project 4, you used *procedural* programming to create a complex number calculator. In this project, you will use *object-oriented* programming (OOP)! You will implement a class `Complex` to represent complex numbers of the form $x + yi$. Then you will use class `Complex` in two separate applications: a piece of fractal art and another complex number calculator. Yes, once you have implemented the `Complex` class, you can use it in many different ways! “Code re-use” is one of the benefits of OOP.

Part A: Java API

Read pages 36–45 of the Savitch text on Classes and Strings. (On `String` methods, just skim—no need to read the methods in detail and definitely do *not* try to memorize them!) `String` is one of many classes in the Java API. Next go to the documentation of the Java API at <http://java.sun.com/j2se/1.5.0/docs/api/> to find the answers to the following questions.

1. What does API stand for? Briefly explain what the Java API is.
2. Take a look at the `Math` class. How many `abs` methods (absolute value) are there? What are the two *fields* in the `Math` class?
3. Take a look at the `System` class. Which method do you use to get the current system time?
4. Take a look at the `Double` class. What does the *field* `MIN_VALUE` store?
5. Take a look at the `String` class. What does method `trim` do?
6. Take a look at the `DecimalFormat` class. How do the pattern characters `0` and `#` differ?

You do not need to read the API documentation in detail to answer the above questions. The objective of this exercise is for you to learn about the API and to learn how to find detailed specifications should you have the need in the future. To answer most of the above questions, you only need to skim through the first pages of the description or tables for each class. Save your answers in a plain text file `api.txt` and submit it to CMS.

Part B (Complex Numbers) will appear in a separate document.