

The Point of This Course

CS 5010 Program Design Paradigms
“Bootcamp”
Lesson 0.1



Learning Objectives

- By the time you complete this lesson, you should be able to :
 - Explain **the point** of the course
 - list the 7 principles for writing beautiful programs
 - list the 6 steps of the design recipe
 - recite some of the slogans that we will use throughout the course.

The Point

1. It's not calculus. Getting the right answer is **NOT ENOUGH**.
2. The goal is to write **BEAUTIFUL PROGRAMS**.
3. A beautiful program is one that is readable, understandable, and modifiable by people.

Your programs should look like this:



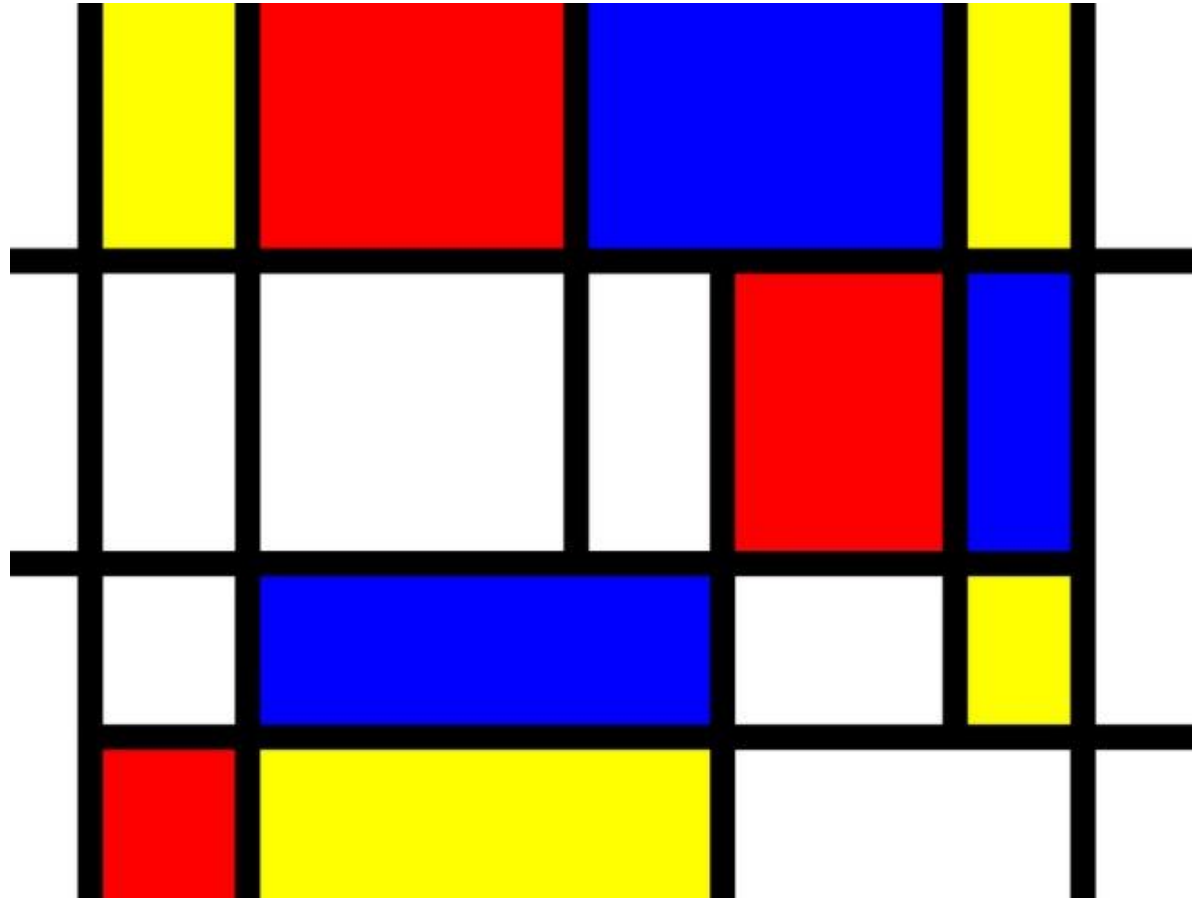
[source](#)

Not like this



[source](#)

Your programs should look like this



[source](#)

Not like this



[source](#)

And never, ever like this



[source](#)

Seven Key Practices for Writing Beautiful Programs

1. Write programs that people can read, understand, and modify.
2. Represent information as data; interpret data as information.
3. Use contracts and purpose statements to specify the intended behavior of your functions and methods.
4. Use invariants to limit your functions' responsibility.
5. Use functions and methods that communicate by passing arguments and returning values
6. Use global side-effects only to share information between distant parts of the program.
7. Use interfaces to limit dependencies between different parts of your program.

The Key Practices in Action

- Everything we do can be traced back to one or more of these key practices.
- We will expand on each of them as we go along.
- Write these down, in your own handwriting. Writing things down will help you remember them.

The Function Design Recipe

- This recipe tells you the *order* in which to attack a programming problem.
- You need to do these steps *in order*:
- If you haven't specified your data, you won't know what your data looks like or what it means.
- You can't write a function that does its job unless you know what its job is.

The Function Design Recipe

1. Data Design
2. Contract and Purpose Statement
3. Examples and Tests
4. Design Strategy
5. Function Definition
6. Program Review

This is important. Write it down, in your own handwriting. Keep it with you at all times. Put it on your mirror. Put it under your pillow. I'm not kidding!

A Few of Our Slogans

- We are also big on slogans. We think they help focus your mind.
- Here are our first few slogans. You should write them down, too, in your own handwriting.
- In fact, whenever you see one of these blue tables, you should assume that this is something important, and you should probably write it down in your own handwriting so you can memorize it.

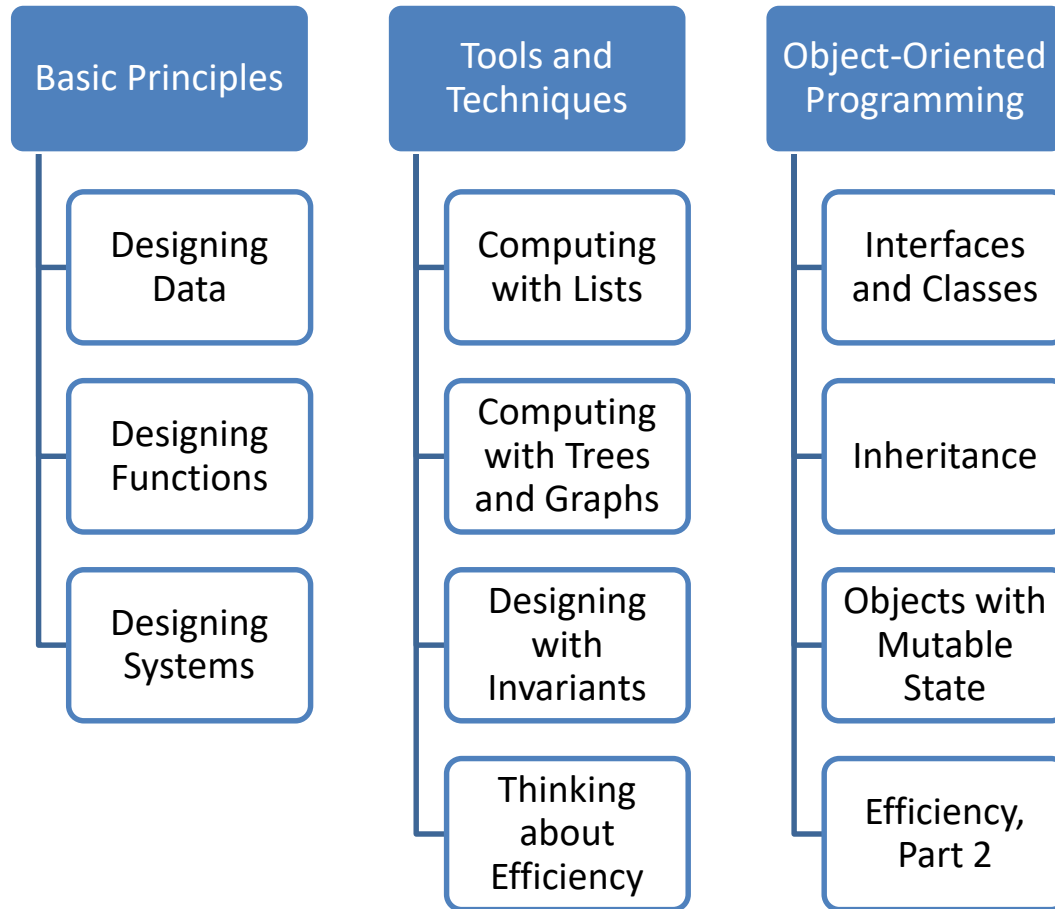
Some Slogans

1. Follow the recipe!
2. You don't understand it until you can give an example.
3. One function, one task.
4. The Shape of the Data Determines the Shape of the Program.
5. Practice makes perfect.

The Course Map

- The course is divided into 3 main units:
 - Basic Principles
 - Tools and Techniques
 - Object-Oriented Programming
- The first unit is taught in Racket; the second in a mixture of Racket and Java; and the third in Java.
- The map on the next slide, which we will show at the beginning of every module, will help you see where you are in the course content.

Course Map



Next Steps

- If you have questions about this lesson, ask them on Piazza
- Go on to the next lesson