

## CSC207.01 2014S, Class 24: Java Generics

---

### Overview

- Preliminaries.
  - Admin.
  - Questions.
- Topics.
  - Homogenous vs. heterogenous collections.
  - Writing general code with type variables.
  - Java's generics.
  - Generic classes, interfaces, and methods.
  - Handling multiple types.
- Lab.

## Preliminaries

### Admin

- Have fun with Earnest!
- Note that I talked a bit about program design at yesterday's extra. The eboard is posted.
- Homework 5 is due March 5.
- The exam makeup is due Sunday at 10:30 p.m. There will be no extensions.
- Extra credit:
  - Harris wellness thingy tonight
  - Galaxy dinner and auction tonight <http://www.galaxyinc.org>
  - Convocation next Wednesday.
  - Grinnell image presentations next Thursday and Friday.

## Questions

*How do we arrive at the invariant for the "Dutch National Flag" problem?*

There's no particular strategy for arriving at invariants. It's like much of algorithm design: You think about the problem for awhile. In the case of invariants, the idea is that once you've *started* to design an algorithm, you have a model for the state of the system. The invariant represents that state.

In this particular case, I expect that midway through the algorithm, I'll have a section of red values, a section of white values, a section of blue values, and a section of unknown values.

*In part D, I'm not sure about "one of which only takes on values of  $x^{2^k}$ ". I think it should be " $x^{(2k)}$ ", not " $x^{2^k}$ ".*

$x^{2^k}$  is  $x^{(2^k)}$  and not  $(x^2)^k$ . And that's what I've written. Exponentiation is right-associative. And I do intend that it be "x to the powers of two". But that may not be the best invariant for you; it's just how I find it useful to think about the problem.

$x_{k+1} = x * x_k$ , for even  $k \geq 0$ . I think you mean "every", not even.\_

While it's true for every  $k$ , I do mean "for even  $k$ ", because I only want this rule to apply to odd exponents.

## **Homogenous vs. heterogenous collections**

## **Writing general code with type variables**

## **Generic classes, interfaces, and methods**

## **Handling multiple types**

Copyright (c) 2013-14 Samuel A. Rebelsky.



This work is licensed under a Creative Commons Attribution 3.0 Unported License. To view a copy of this license, visit <http://creativecommons.org/licenses/by/3.0/> or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.