CIS 194

WELCOME

“TRUST ME, IT WORKS!”

```
public final abstract class BubbleSort {
  public static List<Integer> sort(List<Integer> list) {
    for (int i = 0; i < list.size(); i++) {
      for (int j = i + 1; j < list.size(); j++) {
        if (list.get(i) > list.get(j)) {
          int temp = list.get(i);
          list.set(i, list.get(j));
          list.set(j, temp);
        }
      }
    }
  }
}
```

TOTALLY REAL CODE FOR LAUNCHING ICBM'S

Typical imperative language:

```
shouldLaunchMissiles = false;
if (shouldLaunchMissiles = true) {
  fireTheMissiles();
}
```

... uh oh!

LET'S PLAY: GUESS THE INTENDED USE OF THIS FUNCTION

```
def double(x):
    return 2 * x
```

Maybe `double(5) == 10`? Wrong.

How about `double("hello") == "hellohello"`?

Actually the correct answer is (obviously):
`double([5, "a"]) == [5, "a", 5, "a"]`.

WHY HASKELL?

THE TYPICAL DESCRIPTION

- Functional language
- Pure computation
- Strong static type system
- Lazy evaluation

COULD YOU REPEAT THAT IN ENGLISH, PLEASE?

- Functional language
  - Declarative, expressive code
- Pure computation
  - No side-effects means easier to reason about
- Strong static type system
  - Fewer bugs at runtime, always up-to-date docs
- Lazy evaluation
  - Good performance, infinite data structures

HALLMARKS OF HASKELL

- Rich abstractions
- Combining smaller functions to create complex behavior
- Very little “accidental complexity”[1]

[1] Term coined by Fred Brooks in “No Silver Bullet”

WHY LEARN IT?

- Paragon of FP languages
- Forces you to think differently
- Makes you better programmer in other languages!
- Shows you the future of programming

WHAT IS IT USED FOR?

- Programming language research
- Highly-parallel systems
- Compilers
- **And pretty much anything else**

WHAT ARE WE GOING TO LEARN?

FUNDAMENTALS

- Functions
  - Recursion, composition
- Types
  - Reading types
  - Modeling data
- Higher order functions
  - fold, map

FANCY HASKELL STUFF

- Typeclasses
- Monoids
- Functors
- Monads

Don’t worry if these sound like made-up words to you right now, all will be clear in due time!

REAL WORLD APPLICATIONS

- Backend for a simple web app
- Parsing and interpreting a small language
- Communicating with APIs via HTTP requests

ADMINISTRIVIA

INSTRUCTOR

PALMER PAUL
palmerpa@seas
OH: Sun. 11-1, Rodin Mezz

- CIS Major
- Class of 2020
- Love functional programming
- ~2 years of Haskell experience
- TA'd CIS 120 for 3 semesters

TEACHING ASSISTANTS

ROB ZAJAC

rzajac@seas

OH: Mon. 3-5, TBD

SANJIT KALAPATAPU

sanjitk@seas

OH: Wed. 2-4, TBD

HOMEWORK

- Mostly every week
- Assigned shortly after lecture
- Due at 11:59 PM on Thursday (unless stated otherwise)
- Graded for correctness and style

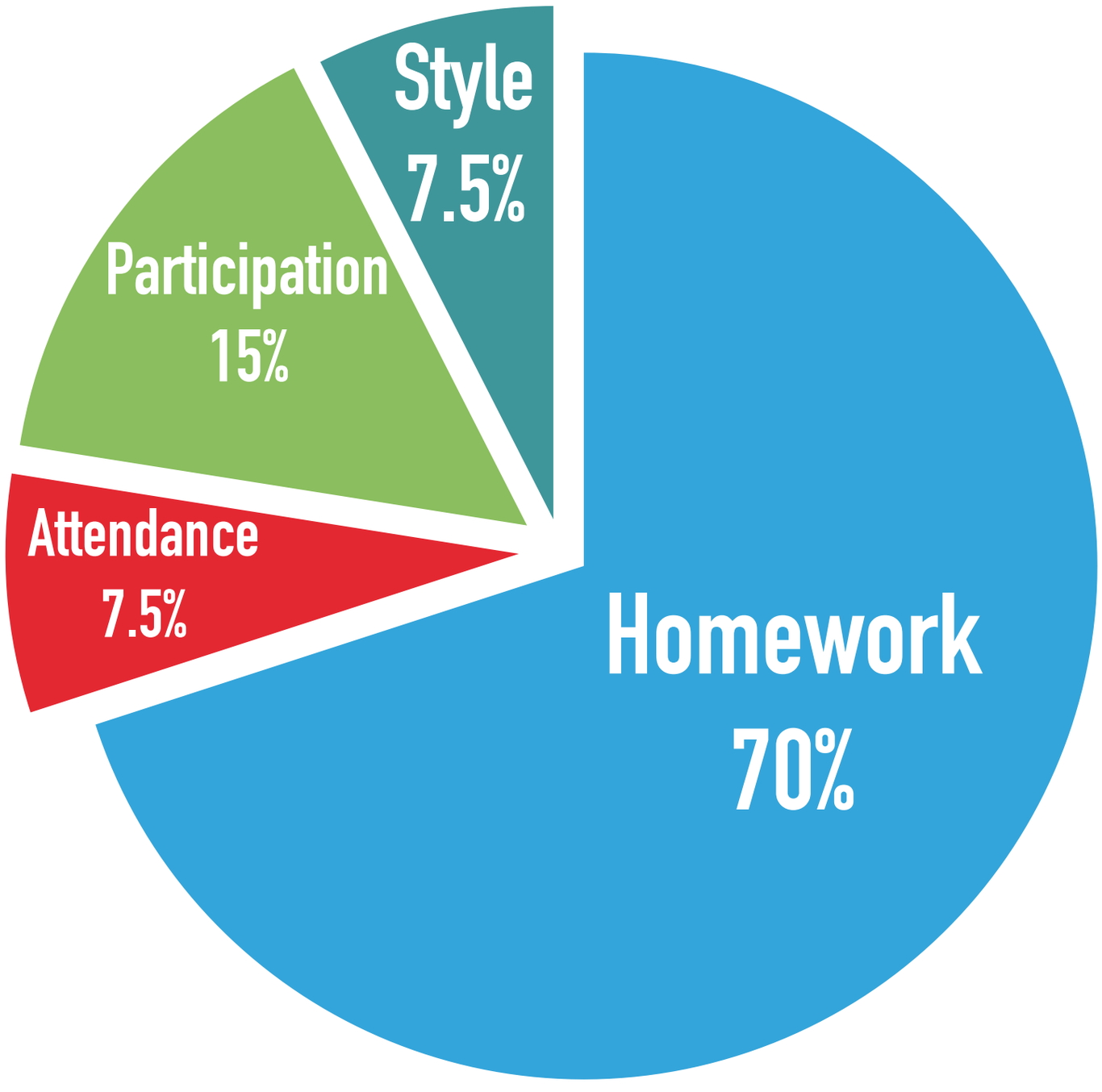
HOMEWORK COLLABORATION

- To be done individually
  - Do NOT talk about specific homework problems
  - But feel free to discuss general concepts
- Restrict online resources
  - GOOD: Piazza, Haskell docs, textbooks, etc.
  - BAD: GitHub, searching for problem, post on StackOverflow
  - When in doubt just ask!

LATE DAYS

- Can re-submit any assignment 24 hours late
- Earn up to 50% extra credit on what you didn't complete
- No further late days*
- Example: I submitted at 10 PM on Thursday and got an 80. When I re-submitting on Friday afternoon, I got a 90. So my final score is an 85.

GRADING



PIAZZA

- Use for all technical questions
- Prefer public posts
- Try to not include code
- Student answers are encouraged

Sign up link: piazza.com/upenn/fall2018/cis194

PIAZZA TEMPLATE

- Clear title
- Question / Problem
- Context
- What have you tried?

Sign up link: piazza.com/upenn/fall2018/cis194

TEXTBOOKS

- *Real World Haskell* by O'Sullivan, Goerzen, and Stewart
- *Learn You a Haskell for Great Good* by Lipovača
- BOTH ARE AVAILABLE FOR FREE ONLINE!

DEMO TIME

INSTALLATION

INSTALL INSTRUCTIONS

- Follow https://www.haskell.org/platform/
- `cabal install HUnit`
- `cabal install hlint`
- Editor
  - Atom with `language-haskell` and `linter-hlint`
  - Or whatever you like best