



**KEEP  
CALM  
AND  
LOVE  
OOP**

# OOP Fundamentals

- Everything is an object
- An object is an **abstract data type** with the addition of **polymorphism** and **inheritance**
- **Encapsulation** is for reducing complexity of a system

# Everything Is An Object

- 42 is an object of type **Integer**
- “Hello world” is an object of type **String**
- Generally, **%TypeName%** is an object of type **Type**
  - But this is rarely used
- Need own type – just define it!

# Abstract Data Type

- Mathematical model for type from domain with similar behavior

# Polymorphism

- From Greek *πολύς*, polys, "many, much" and *μορφή*, morphē, "form, shape"
- Provision of a **single** interface to entities of **different** types
- "If it **looks** like a duck, **swims** like a duck, and **quacks** like a duck, then it **probably** is a duck."



# Inheritance

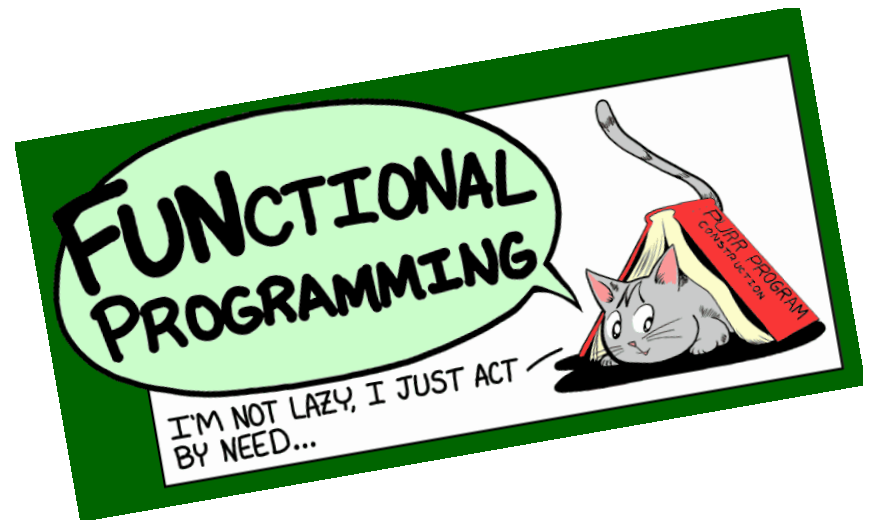
- Class is based on another class
- Inheritance is not a subtyping
  - Subtyping establishes **is-a** relationship
  - Inheritance **reuses implementation**

# Encapsulation

- Encapsulation is the packing of data and functions into a single component
- It [single component] could be
  - Class
  - Module
  - Package
  - Method
  - i.e. **any** single component

# Olympiad Oriented Programming

- Just a joke
- OOP requires a lot of code
  - A lot of time
  - A lot of testing
- FP is a silver bullet?
  - Write less and get more
  - Can be more efficient
    - Lazy evaluation
    - Optimized for recursion





# No, It Isn't

- No support for FP
  - ACM ¼, Python - experiment
- FP requires strong problem description
- OOP sometimes requires less code

