

# LAB Session

- BREAK!
  - Check time. 11:30 must start on assignment
- Watch (lab)
- Do (assignment)
- Sometimes lectures feed into this session
- Other times it doesn't (data science lectures)



POPULATION  
INFORMATICS  
RESEARCH GROUP



# Optional Lab

- TA: Yao Tian
- Tentative: Friday 2-4
- Office hours: Computer Lab
  - Wednesday: 1-3
  - Friday: 4-6



POPULATION  
INFORMATICS  
RESEARCH GROUP



# Programming: Wikipedia

- **Computer programming** (often shortened to **programming**) is the comprehensive process that leads from an original formulation of a computing problem to executable programs. It involves activities such as analysis, understanding, and generically solving such problems resulting in an algorithm, verification of requirements of the algorithm including its correctness and its resource consumption, implementation (or coding) of the algorithm in a target programming language, testing, debugging, and maintaining the source code, implementation of the build system and management of derived artefacts such as machine code of computer programs. The algorithm is often only represented in human-parseable form and reasoned about using logic. Source code is written in one or more programming languages (such as C++, C#, Java, Python, Smalltalk, JavaScript, etc.). The purpose of programming is to find a sequence of instructions that will automate performing a specific task or solve a given problem. The process of programming thus often requires expertise in many different subjects, including knowledge of the application domain, specialized algorithms and formal logic.



# Programming: Wikipedia (cont)

- There is an on-going debate on the extent to which **the writing of programs is an art form, a craft, or an engineering discipline**. In general, good programming is considered to be the measured application of all three, with the **goal of producing an efficient and evolvable software solution (the criteria for "efficient" and "evolvable" vary considerably)**. The discipline differs from many other technical professions in that programmers, in general, do not need to be licensed or pass any standardized (or governmentally regulated) certification tests in order to call themselves "programmers" or even "software engineers." Because the discipline covers many areas, which may or may not include critical applications, it is debatable whether licensing is required for the profession as a whole. In most cases, the discipline is self-governed by the entities which require the programming, and sometimes very strict environments are defined (e.g. United States Air Force use of AdaCore and security clearance). However, representing oneself as a "professional software engineer" without a license from an accredited institution is illegal in many parts of the world.
- Another on-going debate is the extent to which the **programming language used in writing computer programs affects the form that the final program takes**. This debate is analogous to that surrounding the Sapir–Whorf hypothesis in linguistics and cognitive science, which postulates that **a particular spoken language's nature influences the habitual thought of its speakers**. Different language patterns yield different patterns of thought. This idea challenges the possibility of representing the world perfectly with language, because it acknowledges that the **mechanisms of any language condition the thoughts of its speaker community**.

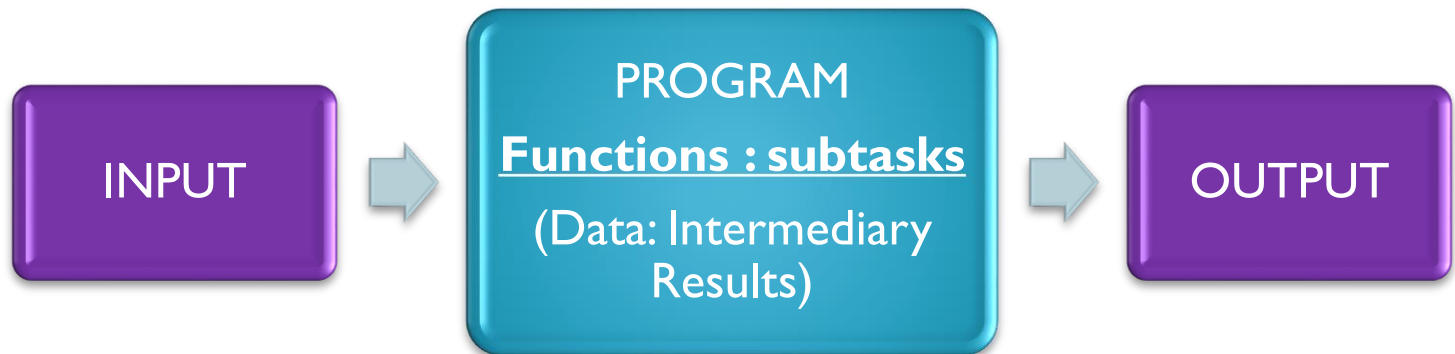


# Programming for us

- Code/language = SAS
- Reusable Code
  - vs point & click
- Software Engineering
  - Project management
  - Build a process (sequences of codes/data) that can be efficiently managed
  - Need to be able to pick up after 6 months
- Very disciplined activity
  - Must keep to rules

# Programming: Hye-Chung Kum

- Take INPUT and turn it into OUTPUT
  - OUTPUT : Know what you want/goal
  - INPUT : figure out what you have to work with
- Change what you have (INPUT) to what you need (OUTPUT)
  - Break up the problems into small subproblems
  - Intermediate results (scrap paper)
  - Use functions to calculate the intermediary results



# Working environment

- Computer lab
  - Must save out files to/from dropbox or USB
  - Recommend running directly from C:/
  - Recommend using the same computer
    - No guarantee on files
- On your laptop
  - Install before next week
- On your desktop
  - If home PC: Install before next week
  - If office: figure out software before next week
  - Figure out how to remote into from the lab
- Virtual Lab



# Hello World

- SAS
  - How to start
  - Saving out log, lst
- Text editors: window into communicating with computers
- If time permits, go over code in the assignment





# Resources

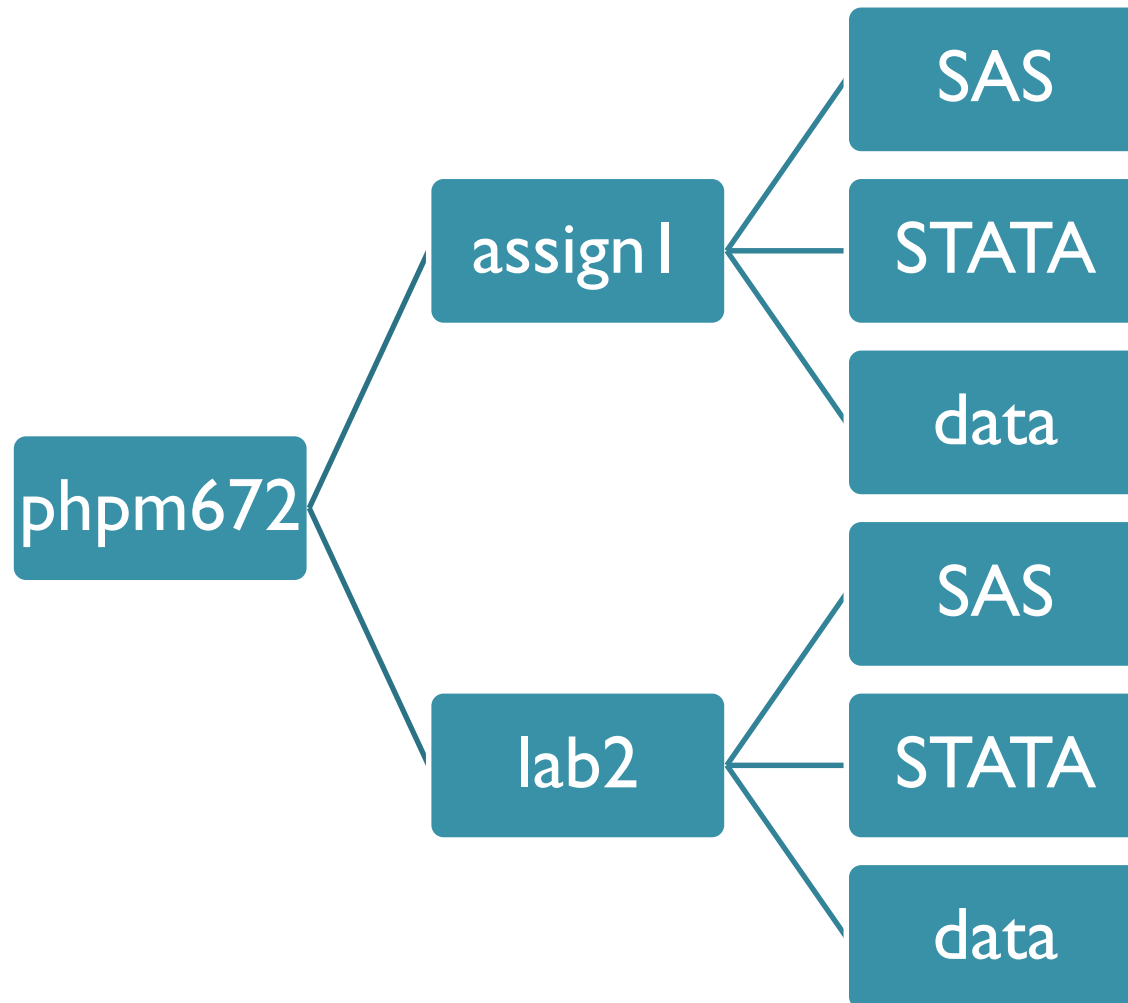
- Population informatics website

# Programming

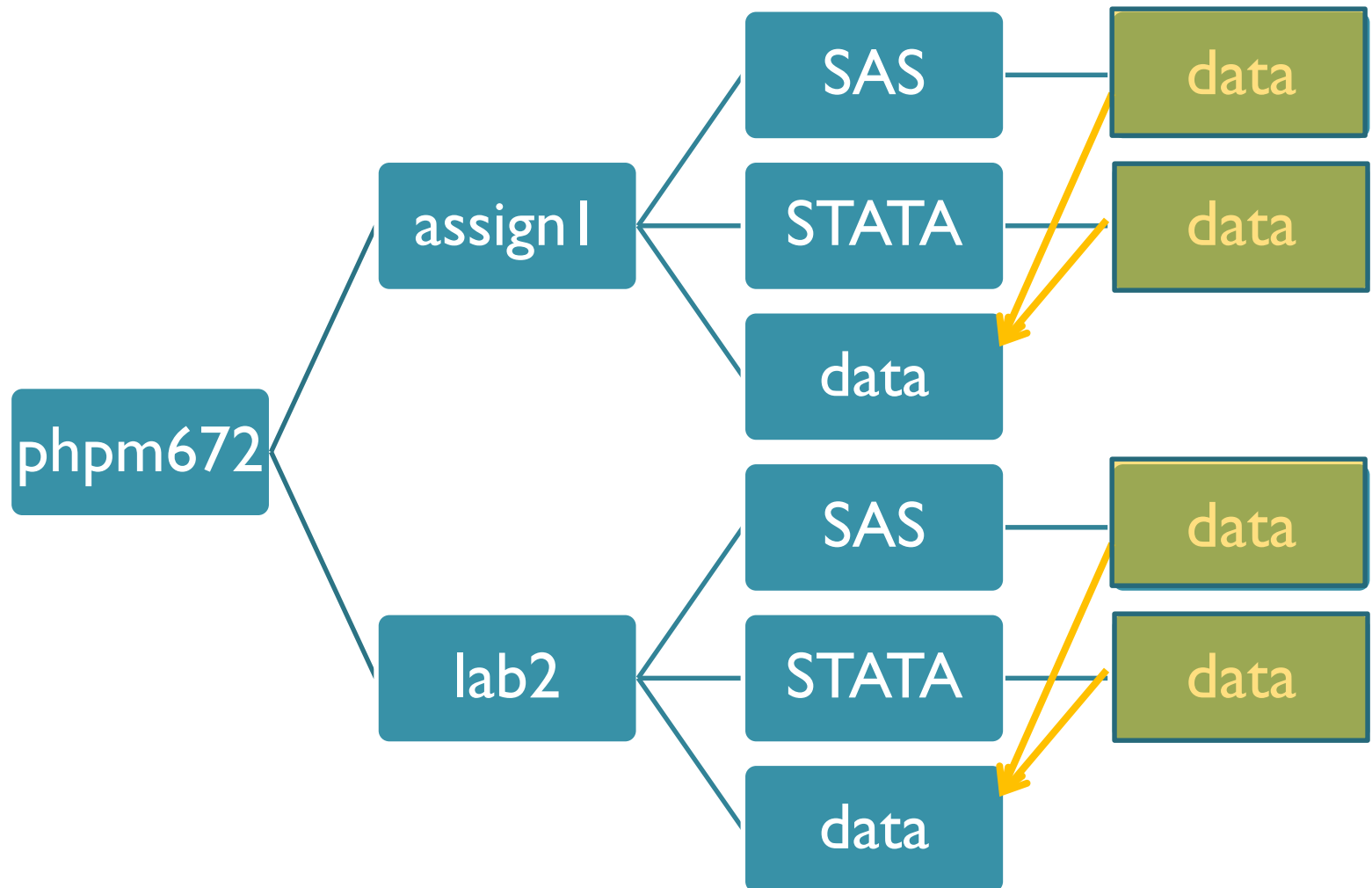
- Know your computer
  - Do you know exactly where everything is?
  - Be the boss, stay in charge
- File management:
  - logical organization that makes sense to you
  - More than one logical organization
    - Short cuts
    - Symbolic links (ln -s)
  - Not too deep, but enough to keep organized
- File extensions
  - Tells your computer what the files are for



# Physical structure



# Logical structure



# Remember

- Read
  - Read recommended modules in the assignment
- Practice
  - Try using SAS
  - Try examples in the modules you are reading
- Assignment #1
  - Due 11:59pm on Jan 25 (Monday)