Introduction
Software Development

Spring 2025

Dr. William Gregory Johnson
Department of Computer Science
Georgia State University

Georgia State University | COLLEGE OF ARTS & SCIENCES

Introduction

- Course Specifics
  - Learning outcomes
  - Structure and format
  - Testing and labs
  - GTAs (lab assistants in this course)
  - Comparison of software development and software engineering
  - Syllabus
  - Weekly course schedule
- A ‘formal course’ specific to software development
  - You all are 2nd year for this new course
  - We have a courseware for career readiness in CS that will be part of this course
  - Many overlaps to CSc4350 (Software Engineering), CSc1302 (java)
  - More knowledge of programming concepts using OOP with java

Introduction

- Learning outcomes
  - Describe essential elements of software architecture, views, and styles
  - Understand proper use of OOP principles: objects/classes, modularity, exceptions, and assertions, S.O.L.I.D.
  - Construct formal reports and demonstrations of working software
  - Describe and understand refactoring versus re-engineering system software
  - Describe and understand inheritance, overloading, user interfaces, callbacks, separation of model from view/control
  - Understand use of basic linear data structures for storage/manipulation of large and complex information and specific structured data types, including arrays, vectors
  - Demonstrate the correct function of a program by developing a plan to verify

Introduction

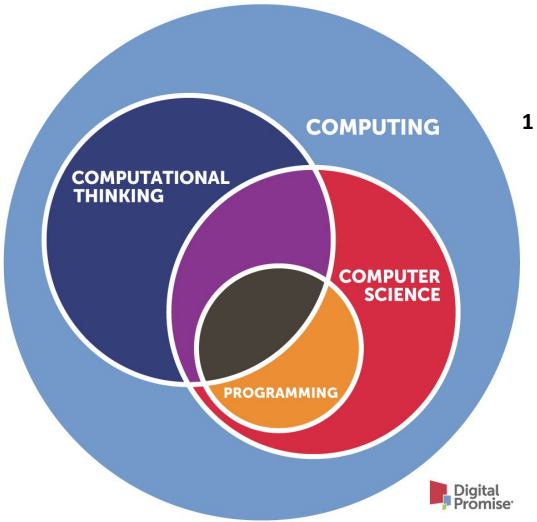
- Structure and format: Labs and Testing
  - Lecture each class, 60 to 90 minutes
  - In-class lab exercise given on entire class time or after brief (<30min) lecture
  - Brief overview and explanation of lab, then exercise given
    - Will be due at a specific time (upload to iCollege)
  - Regular exams given in class, with paper, no scantron needed
    - Several multiple choice (45 to 65 points)
    - A few complex problem solving
    - Three and final exam
  - No dropping of lowest marks on the exams or labs
  - Gradescope used for exam grading, feedback, and return
  - End of semester project

Introduction

- Software development
  - Using some proven methodology to deliver a software system
  - Software Development Life Cycle (SDLC) or Application Life Cycle (‘App Life Cycle’)
  - Activities at a minimum:
    - Discovery of software system requirements
    - Choose a programming language (appropriate for solution and users)
    - Begin the implementation (code->test->refactor->repeat)
    - Evolution (upgrades and maintenance)
  - Putting these together is an abstraction for quantifiable work to create a solution
    - Remember your training: simplify, identify patterns, design an algorithm, generalize for other uses
    - What does this remind you of from computer science?

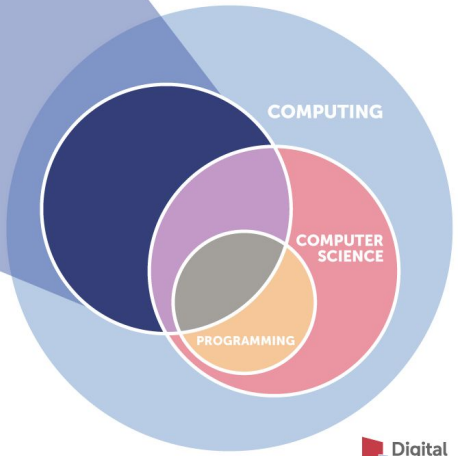
Introduction

Software development = ?
Computational thinking



1

1



[1] https://digitalpromise.org/initiative/computational-thinking/computational-thinking-for-next-generation-science/what-is-computational-thinking/

Introduction

- **Software Development**
  - refers to the process of designing, creating, testing, and maintaining software applications or systems
  - involves a series of steps that begin with understanding the requirements and objectives of the software, followed by designing the architecture and any potential user interfaces
  - development phase typically involves writing code in a programming language, implementing algorithms, and integrating various components
- **Software Developer**
  - are typically responsible for the implementation of specific parts of a software system
  - are more likely to be involved in the day-to-day coding and testing of software

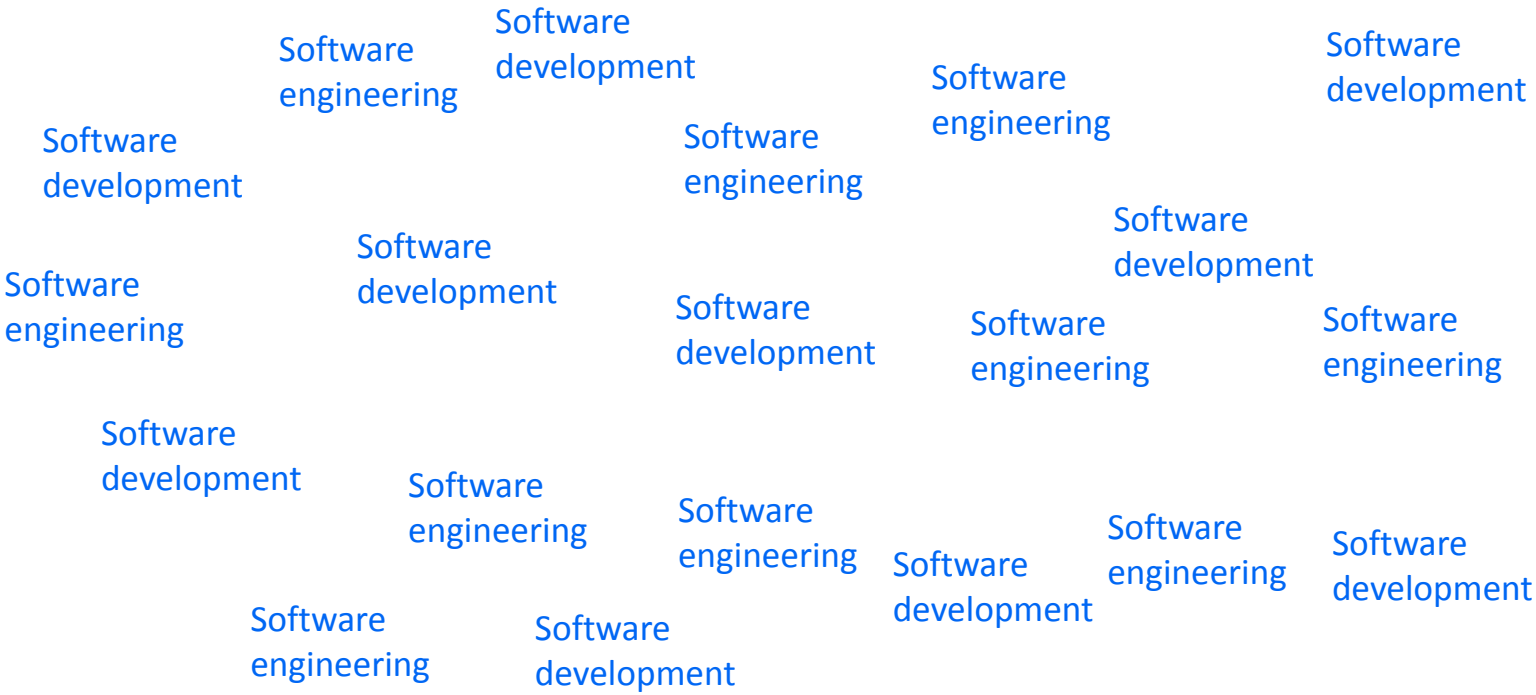
Introduction

- Software engineering
  - Using proven set of activities and producible documents to develop a software system
  - Activities at a minimum:
    - Specifications
    - Design and implementation
    - Validation and verification
    - Evolution (upgrades and maintenance)
  - Support tools are integral
    - Advanced IDE
    - Git (version control)
    - Task tracking (Jira)
  - Using theories, methods, models along with tools defines professional software development
    - Agile
    - Waterfall
    - DevOps (continuous integration / continuous deployment)

Introduction

- Software Engineering
  - is a broader discipline involving entire software creation from idea to production deployment
  - applies engineering principles for design, development, testing, maintenance
  - involves numerous persons (roles) to achieve a solution in an App Life Cycle
- Software Engineer
  - are typically responsible for the overall architecture of a software system.
  - are more likely to be involved in the planning and management of software projects.
  - typically have a broader range of skills, including engineering, management, and design.

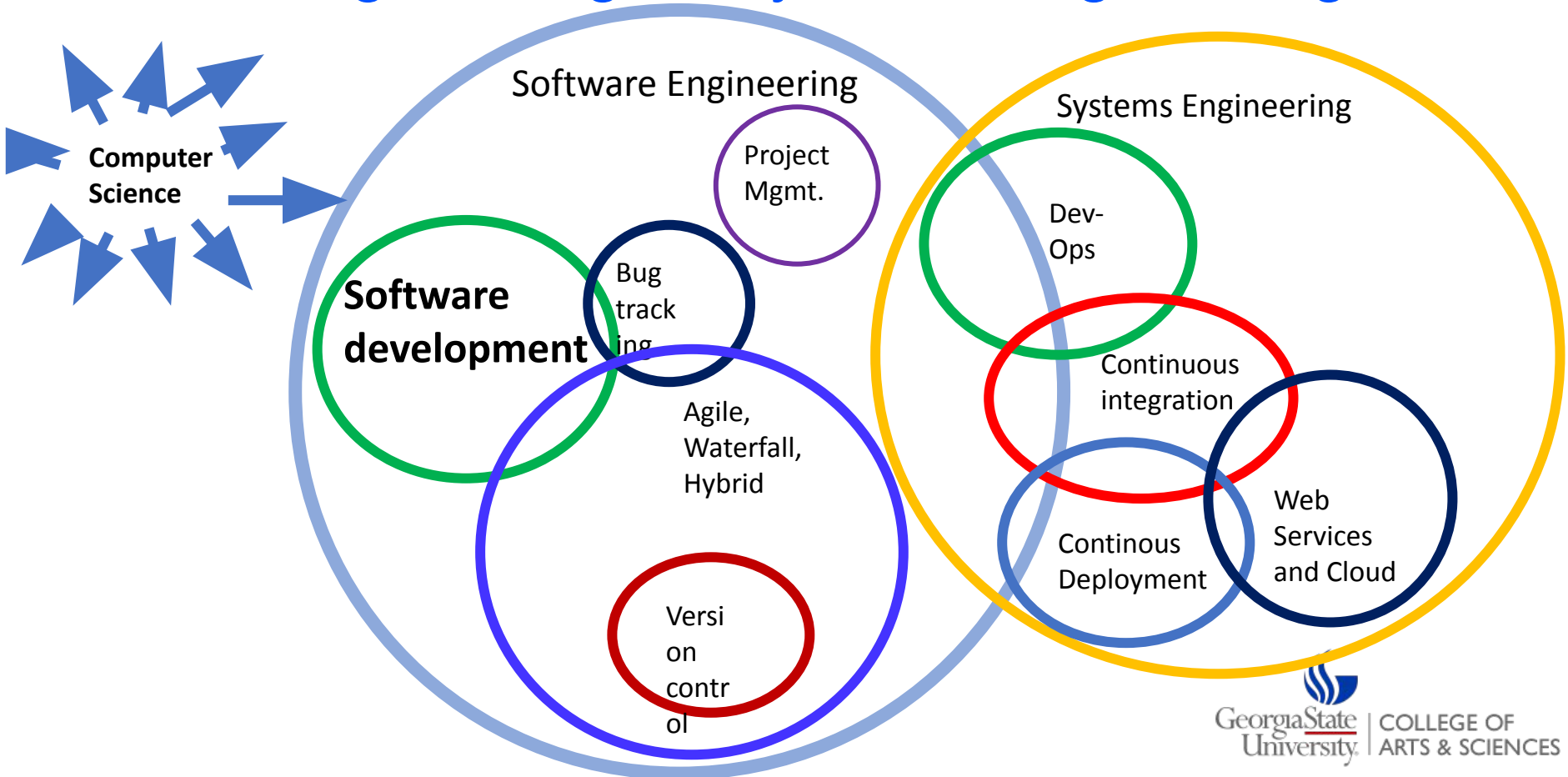
So, what is the difference?



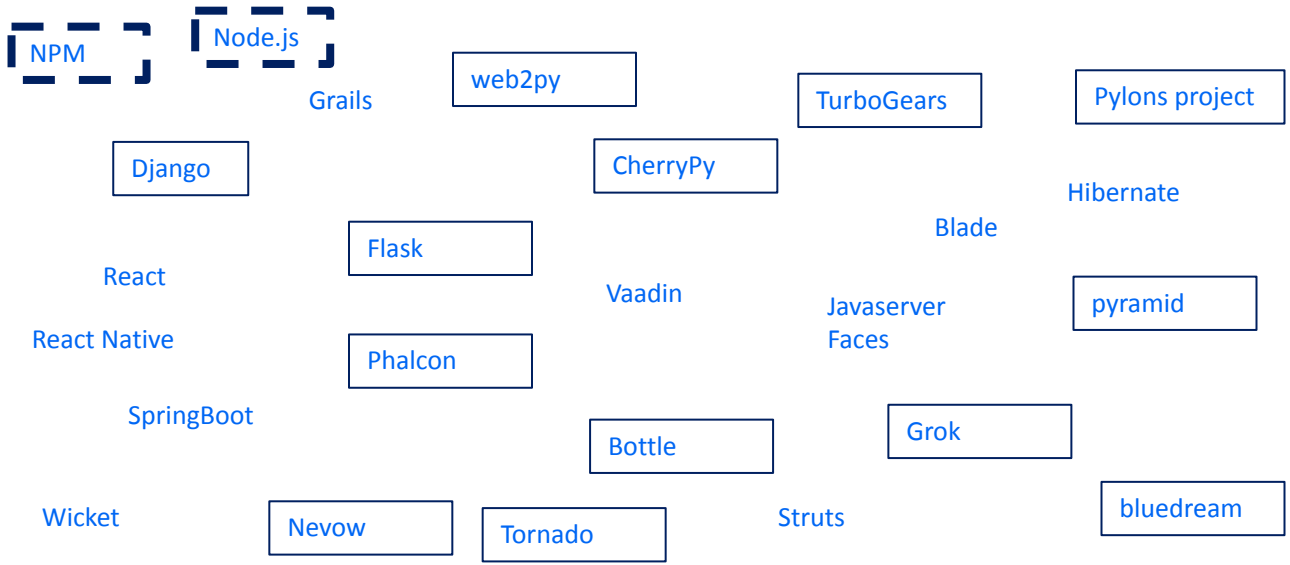
So, what *are* the *differences*?

Georgia State University | COLLEGE OF ARTS & SCIENCES

Software Engineering vs Systems Engineering



Micro-frameworks for Java, JavaScript, and Python [1,2]



So, what to choose?

*python in boxes

1. https://radixweb.com/blog/best-java-frameworks
2. https://www.g2.com/categories/python-web-frameworks

Contrasts and Similarities

- Software Development
  - More emphasis on component design, optimization, standards of coding
  - Testing and ethics should be embedded in all coding models and designs
  - Not involved with overall architecture of a system or multi-component interoperability and configuration
- Software Engineering
  - Broader scope of software creation activities:
    - Project or Product manager
    - Scrum master (depending on model)
    - System architect
    - Senior developer
    - Systems analyst
  - Based on engineering principles in sequenced work to gather requirements, design, implement, test, refactor

Syllabus

- Course specifics
  - Dynamic class content, so open to evaluation for improvement, optimization
  - A survey or two
  - Not a class to learn full-stack tech and advanced programming
  - Very deep into OOP using java
  - SQL, RDBMS tools, Visual Studio Code
  - Many small labs, 3 exams, final exam
  - One project TBD if multiple person teams or not

Syllabus

Syllabus

Syllabus

Syllabus

Syllabus
Schedule

Discord Server for Help

https://discord.gg/umCWuKTzGw

- Chandra: Wednesday 10:00 a.m. – 12:00 p.m.
- Shishir: Tuesday 10:00 a.m. - 12:00 p.m.
- Satvik: Friday 3:00p.m. - 5:00 p.m.
- Krishnanjali: Thursday 10:00 a.m. – 12:00 p.m.

The end,
Thank you.