CSCE 240: Advanced Programming Techniques (Honors)

Lecture 1: Introduction

PROF. BIPLAV SRIVASTAVA, AI INSTITUTE

10TH JANUARY 2023

Carolinian Creed: "I will practice personal and academic integrity."

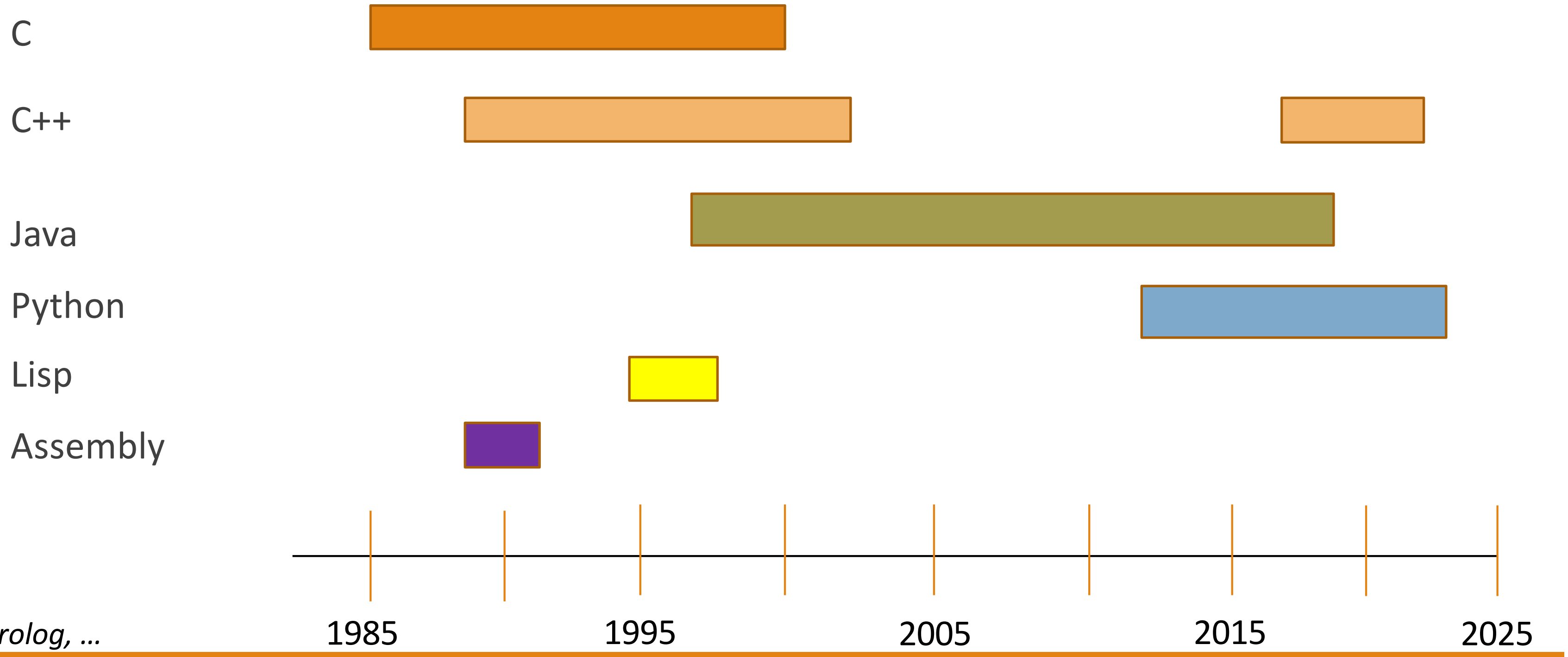
Credits: Some material reused with permission of Dr. Jeremy Lewis. Others used as cited with thanks.

Organization of Lecture 1

- Introduction Section
  - Instructor introduction and course logistics
- Main Section
  - Programming and languages
  - Getting started: the “Hello World !” program
  - Topics, Home works and course project
  - Additional Tasks
- Concluding Section
  - About next lecture – Lecture 2
  - Ask me anything

Introduction Section

Personal Programming Language Journey* (35+ years)

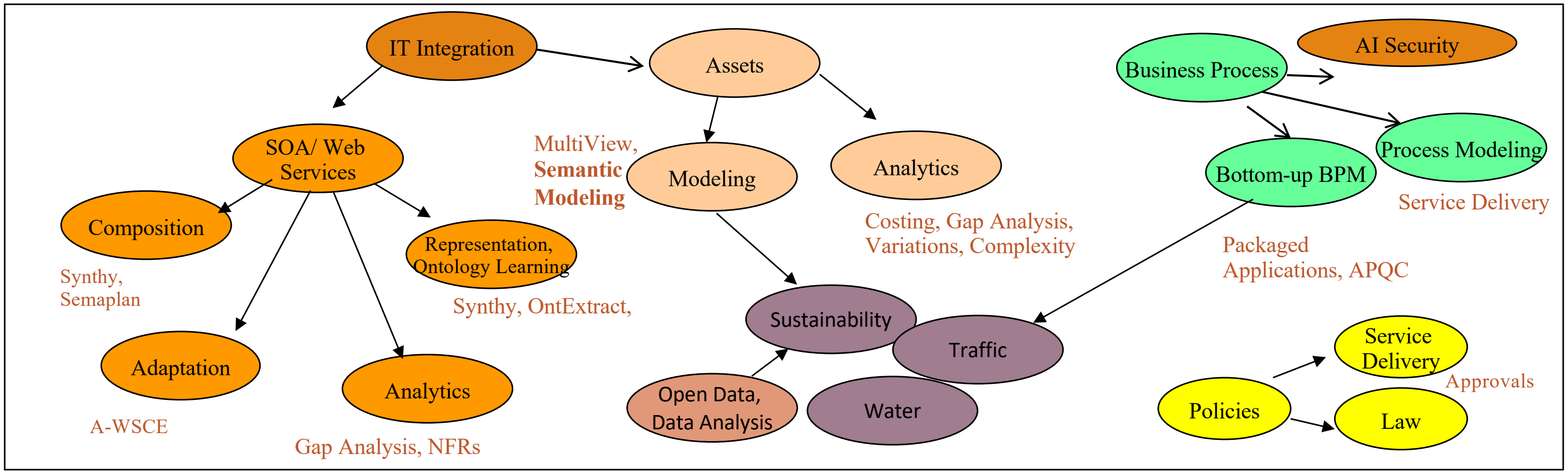


**Excluded:* *Javascript, Prolog, …*

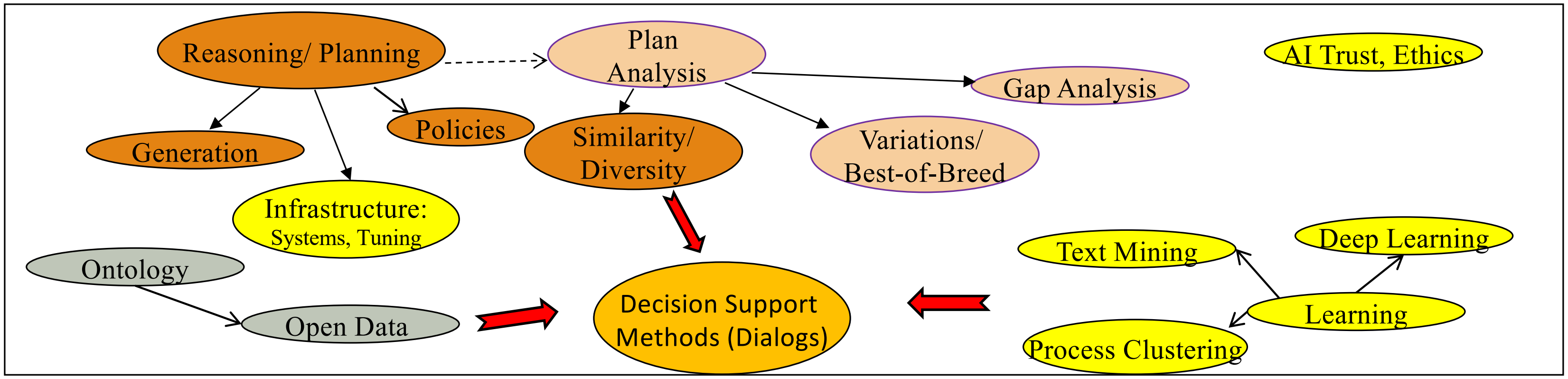
Keywords: AI, Services, Sustainability
Papers: 190+ refereed; ~5200 references
Patents: 68 (US issued); 4 sole inventions

BIPLAV SRIVASTAVA Research Snapshot (1989-2023)

The Space of AI Applications Explored



The Space of AI Techniques Used



About the Honors Program, Students

- Honors program expectations
- Students quick survey about programming experience

Course Logistics

Administrative Information

- HNRS: Advanced Programming Tech. CSCE240-H01-SPRING-2023
- Meeting Time: TuTh 8:30—9:45 AM
- Class methods
  - In-class: Swearingen Engr Ctr | Room 2A19
  - Material available afterwards on Blackboard (hence, Asynchronous Online)
- Instructor: Biplav Srivastava, Ph.D.
  email: biplav.s@sc.edu
  office: AI Institute, Room 515, 1112 Greene St., Columbia, 29028
  office hours: By Appointment in-person or Blackboard (11:30 am - 12:30 pm), M and W
- Websites
  - Course: Code and slides - https://github.com/biplav-s/course-adv-proglang-s23
  - Details: https://sites.google.com/site/biplavsrivastava/teaching/csce-240-advanced-programming-techniques

CSCE 240 - Advanced Programming Techniques (3 Credits)
Pointers; memory management; advanced programming language structures: operator overloading, iterators, multiple inheritance, polymorphism, templates, virtual functions; Unix programming environment.
Prerequisites: CSCE 215, C or better in CSCE 146.

Learning Objectives

- Develop language-independent understanding of programming concepts by being exposed to multiple languages (C++, Java, Python)
- Independently design and implement programs in multiple language of choices (C++, Java or Python based on choice) in a Unix environment
- Demonstrate mastery of pointers, iterators, memory management including object creation and destruction, and parameter passing in C++
- Demonstrate mastery of object-oriented programming concepts including: inheritance, polymorphism, operator overloading, template functions and classes, and the use of STL containers.
- Develop object-oriented models using UML
- Able to work in programming teams with code review and walk throughs
- Solve practical problems that matter

Books and Resources: C/C++

- C/C++
  - **(Authoritative)** Brian Kernighan and Dennis Ritchie, The C Programming Language, https://en.wikipedia.org/wiki/The_C_Programming_Language
  - **(Authoritative)** Bjarne Stroustrup
    - The Annotated C++ manual, https://www.stroustrup.com/arm.html
    - The C++ Programming Language (4th Edition), Addison-Wesley ISBN 978-0321563842. May 2013, https://www.stroustrup.com/C++.html
  - Walter Savitch, Absolute C++ 6th ed., Pearson, 2016
  - Free books
    - C++ Essentials, Sharam Hekmat, https://freecomputerbooks.com/Cpp-Essentials.html
    - Fundamentals of C++ Programming , by Richard L. Halterman https://archive.org/details/2018FundamentalsOfCppProgramming/page/n333/mode/2up
    - C++ Today, https://www.jetbrains.com/cpp/cpp-today-oreilly/

Books and Resources: Java, Python

- Java
  - **(Authoritative)** The Java Programming Language, 4th Edition 4th Edition by Ken Arnold, James Gosling, David Holmes, ISBN-13: 978-0321349804
  - Effective Java - 3rd Edition, by Joshua Bloch, ISBN-13: 978-0134685991
  - Free books
    - Essential Java, by Krzysztof Kowalczyk (HTML), https://www.programming-books.io/essential/java/
    - Teach Yourself Java in 21 days, https://cs.cmu.edu/afs/cs.cmu.edu/user/gchen/www/download/java/LearnJava.pdf
- Python
  - **(Authoritative)** https://docs.python.org/3/tutorial/
  - Free books
    - Fundamentals of Python Programming, Richard L. Halterman, https://freecomputerbooks.com/Fundamentals-of-Python-Programming-by-Richard-Halterman.html
    - Think Python, Allen Downey, https://greenteapress.com/wp/think-python-2e/

Main Section

Programming and Languages

Programming – Why and What's New

Why

- Control a system (hardware, software). With
  - Performance – effective, efficient
  - Robustness – e.g., handle variation in inputs
  - Maintenance – manage change in system easily
  - Cost – low
- Changes in programming languages and systems happen continuously

What's New

- Mainstream programming
  - Focus on maintenance and low cost
  - Python, Java
- Specialized programming (e.g., military, games, IoT)
  - Focus on performance
  - C++/C, Assembly
- Automatic programming - new
  - Code generation (Github CoPilot, ChatGPT)

Programming – How You Approach Coding

- Software engineering
  - Requirements
  - Specification
  - Design
  - Coding
  - Testing
- Development in teams
- Communication with all stakeholders
- Meeting project objectives

Languages – How You Conduct Coding

- Language choice
  - Coding convention
  - Code organization
  - Tool choices
  - Coding process
  - Syntax
  - Testing process
- Code maintenance
  - Releases
  - Bug fixing

Getting started:

The “Hello World !” program

C/ C++ - Setup

- Using native command line
  - https://www.tutorialspoint.com/cprogramming/c_environment_setup.htm

- Using IDE
  - Eclipse: https://www.softwaretestinghelp.com/eclipse-for-cpp/

Java - Setup

- Using native command line
  - https://www.tutorialspoint.com/java/java_environment_setup.htm

- Using IDE
  - Eclipse: https://courses.cs.washington.edu/courses/cse373/18au/resources/eclipse-setup.html

Python - Setup

- Using native command line
  - https://wiki.python.org/moin/BeginnersGuide/Download

- Using IDE
  - Eclipse: https://www.ics.uci.edu/~pattis/common/handouts/introtopythonineclipse/
  - PyCharm: https://www.jetbrains.com/help/pycharm/quick-start-guide.html

Topics, Home Work, Project

Topics to Cover

- Input and output
- Pointers
- Iterators
- Memory management including object creation and destruction
- Parameter passing
- Object-oriented programming concepts including: inheritance, polymorphism, operator overloading, template functions and classes, and the use of STL (standard template library) containers.
- Develop / communicate object-oriented models using UML

Teaching Philosophy and Evaluation

- Learning under controlled, supervised environment (“spoon feeding”)
  - Nature
    - Going by topics
    - Assessment by strict rubrics
      - Home work
      - Quizzes
  - Pros: easy to follow by students (especially by non-serious ones)
  - Cons: problems are disparate, mundane
- Freedom with responsibility
  - Nature
    - Solving meaningful societal problems; applying concepts learnt
    - Freedom to choose language, concepts and algorithms
    - Assessment by impact and effort
      - Projects
  - Pros: learn concepts by doing, better job prospects
  - Cons: pro-active effort needed by everyone

Home Work

- Home works will be testing content taught in class

Course Project – Assembling of Prog. Assignments

- **Project**: Develop collaborative assistants (chatbots) that offer innovative and ethical solutions to real-world problems !

(Based on competition - https://sites.google.com/view/casy-2-0-track1/contest)

- Specifically, the project will be building a chatbot that can answer questions about a South Carolina member of state legislature from: https://www.scstatehouse.gov/member.php?chamber=H

- Each student will choose a district (from 122 available).

- Programming assignment programs will: (1) extract data from the district, (2) process it, (3) make content available in a command-line interface, (4) handle any user query and (5) report on interaction statistics.

Code Sharing and Review

- All code will be put on student’s personal GitHub account in a repository named: **csce-240-02-spring2022-programs**
- Repository will be shared with instructor (GitHub: biplav-s) and TA.
- Homework assignments will be peer-reviewed in class. Not graded but class activity (doing home assignments, peer reviewing and testing) will count towards overall grade
- Programing assignments and project will be reviewed by TA and instructor only; select projects will be shared with class with students’ permission

Student Assessment

A	=	[900-1000]
B+	=	[850-899]
B	=	[800-849]
C+	=	[750-799]
C	=	[700-749]
D+	=	[650-699]
D	=	[600-649]
F	=	[0-599]

Tests	**1000 points**
• Course Project: programming assign.(5) and report, in-class presentation	600 points
• Class Participation and Home Work	200 points
• Quizzes and Exams	200 points
Total	1000 points

Course Logistics

Additional Tasks

- Setup GitHub repository
- Extend “Hello World” programs to read an argument from command line, concatenate to “Hello World” and print it

Concluding Section

Lecture 1: Concluding Comments

- We discussed course aims
  - Learn programming techniques
  - C/C++ will be the “mother language”
  - Choose one or more languages to have multi-lingual learning
- Learn important programming concepts
- Learn in real-world setting, i.e., with others
- Solve real-world problems

About Next Lecture – Lecture 2

Lecture 2: Experience with Development Environments

- Review Hello World
- Implement Read/ Write
- Implement sorting of numbers
- Peer code review and testing