CSCE 240: Advanced Programming Techniques

Lecture 2: Experience with Development Environments

PROF. BIPLAV SRIVASTAVA, AI INSTITUTE

12TH 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 2

- Introduction Section
  - Recap of Lecture 1
  - Slack account
- Main Section
  - Review additional tasks: GitHub setup, Enhanced Hello World!
  - Concept: Pointers
  - Concept: Iteration
  - Numeric processing: Quicksort
  - Team activity: code review, testing
- Concluding Section
  - About next lecture – Lecture 3
  - Ask me anything

Introduction Section

Recap of Lecture 1

- We discussed course aims
  - Learn programming techniques
  - **C/C++ will be the "mother language": everyone should know**
  - **Choose one or more languages to have multi-lingual learning: Java or Python preferred**
- Learn important programming concepts
- Learn in real-world setting, i.e., with others
- Solve real-world problems

Technology to Connect

- Slack: for clarifications, quick questions, connect
  - Invitation emails sent
- Email: for longer messages, planned discussions, important communication

Main Section

Review: GitHub

- GitHub basics
  - Sharing code, replicating others work
  - https://docs.github.com/en/get-started/quickstart/hello-world
- Course GitHub
  - https://github.com/biplav-s/course-adv-proglang
- Share your private repo on “Students and Code Links” spreadsheet
  - Name the repo: CSCE240H-Spring2023-<your-first-name>. Example: CSCE240H-Spring2023-Bob
  - Additionally
    - Put a readme file (Readme.md) and give you full name
    - Create a sub-folder called “home-works”. Future home works will be here
    - Create a sub-folder called “project”. Your project will reside here.

Review: Enhanced Hello World

- C++, Java, Python

Crash Course on Computer Organization

- Architecture and Hardware
  - Components: memory, hard-disk, I/O, CPU
  - Working of a computer
- Programming level
  - Variables, locations and values
- Compilers/ Interpreters
  - Connecting program (software) to hardware

Concept: Pointers

- Pointers refer to accessing and manipulating location of variables
  - a = 12 // variable is a, value is 12
  - b = &a // b has the address of a, i.e., 0 here. It is called a pointer
  - c = a // c has the value of a, i.e., 12
  - d = *b // will refer to a. That is, d will be equal to value pointed by b, i.e., 12

Variable	Location	Value
a	0	12
b	4	0
c	8	

Reference: https://www.cplusplus.com/doc/tutorial/pointers/

Pointers in Languages

- C++: fully supported
  - "A pointer is a variable that stores a memory address, for the purpose of acting as an alias to what is stored at that address."
  - Pointer arithmetic
  - Arguments of functions can be passed by value or by pointers
- Java, Python: references
  - "A reference is a variable that refers to something else and can be used as an alias for that something else."
  - When a variables is initialized to another variable, references are passed.
  - No pointer arithmetic by programmer

Reference:

- https://nickmccullum.com/python-pointers/#why-dont-pointers-exist-in-python
- https://www.geeksforgeeks.org/is-there-any-concept-of-pointers-in-java/

Programming Exercise – C++

- Write a function, ***addNumbers()***, with two arguments containing numbers
  - Adds the two numbers
  - Returns the sum

- Write a function, ***addNumbersAtFirstLocation***(), with two arguments containing pointers to variables
  - Adds the two numbers
  - Updates the sum at the location of the first variable

Review With Peers

- Code walk through
- Unit testing

Concept: Iteration

- For – number of iterations is known
  - **for**(initial condition; end condition; update action) {
    //body of the for loop
    }
  - **for**(i = 0; i < 10; i++) {
    cout << i << endl; // print from 0 to 9
    }
- While – number of iterations is unknown
  - **while**(condition) {
    // body of while loop
    }
  - **while**(true) {
    ; // Do nothing -- infinite loop
    }

Numeric Processing

- Problem: Sort numbers
  - **Requirement**: arrange a given sequence of numbers into a sorted (e.g., ascending order)
  - **Specification**:
    - Input: any sequence of numbers with n=0 or more integers
    - Output: a sequence of same length as input but where for i=0..n-2, a[i] <= a{i+1]
  - Design
    - What functions to have?
  - Coding
  - Testing
    - Input is empty string
    - Input has one number
    - Input has two or more numbers
    - Input has all numbers of same value

Illustration: C++

- Usage of pointers
- Passing inputs and outputs is cumbersome
- Notice length of code

Illustration: Java

- Inputs passed as arguments
- Notice smaller length of code

Illustration: Python

- Inputs passed as arguments
- Notice smaller length of code

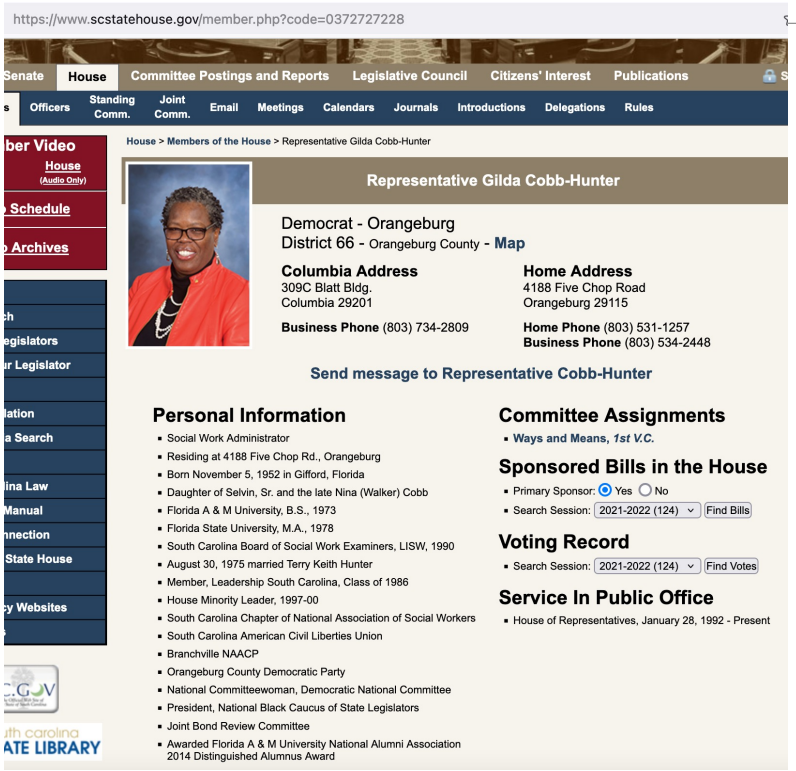
Discussion: Course Project

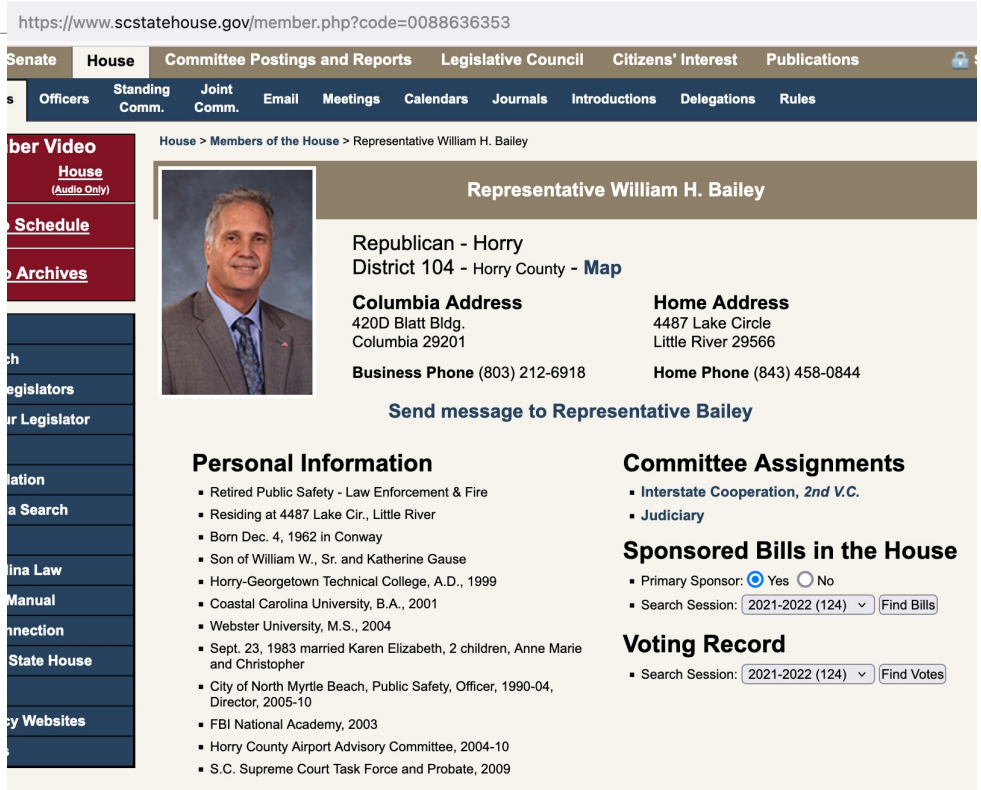
Reference Spring 2022 Course Project – Assembling of Prog. Assignments

- **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.

Problem Scale

- Text and images
- Static and dynamic content; direct and indirect content
- Semi-structured and unstructured content





Discussion: Nature and Simplifications

- Once you select a district, the elected legislator is fixed.
- Some simplifications
  - Download local copy v/s web query
  - Handle static content first
  - Handle a subset of content
  - Have default handling for questions the chatbot does not understand
- Do project in a language you are most comfortable with
- Use all advanced programming concepts to simplify coding

Course Project – Building and Assembling of Prog. Assignments in Health

- **Project**: Develop collaborative assistants (chatbots) that offer useful information about diseases
- Specifically, use the CDC dataset on diseases at: https://wwwnc.cdc.gov/travel/diseases.
  - For polio, it is: https://wwwnc.cdc.gov/travel/diseases/poliomyelitis
  - Each student will choose two diseases (from 47 available).
  - Each student will also use data about the disease from WebMD. Example for polio - https://www.webmd.com/children/what-is-polio
  - Programming assignment programs will: (1) extract data about a disease from two sites, (2) process it, (3) make content available in a command-line interface, (4) handle any user query and (5) report on interaction statistics.

Concluding Section

Lecture 2: Concluding Comments

- We discussed the concepts of pointers and references
- We discussed the concept of iteration
  - For and while are most common
  - Others available (like do-while) but not that helpful in practice

- Looked at enhanced “Hello World”
- Looked at numeric processing – quick sort
- Discussed projects

Additional Tasks

- Implement sorting in C++
- Implement sorting in another language (Java or Python)
- Add code on personal GitHub
- Update Instructor on slack

About Next Lecture – Lecture 3

Lecture 3: I/O

- Overview of streams, file processing
- Error handling
- Printing values