CSCE 240: Advanced Programming Techniques
Lecture 25: Project Reuse Presentations, HW6, Prep Topics for Quiz2

PROF. BIPLAV SRIVASTAVA, AI INSTITUTE

11TH APRIL 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 25

- Introduction Section
  - Recap of Lecture 24
  - News / announcements / clarifications

- Main Section
  - Task: HW 6 – review
  - Project – Guidelines on submission and presentation
  - Review of concepts

- Concluding Section
  - About next lecture – Lecture 26
  - Ask me anything

Introduction Section

Recap of Lecture 24

- Project assignments
  - Programming practices by students in PA#4
  - Reviewed PA #5

- We discussed
  - AI / ML/ DL

Main Section

Home Work 6 (Peer Review)

Due Tuesday, April 9, 2024

Home Work (#6) – C++ - Understand Code Optimization

- Consider BubbleSort algorithm for sorting (shown on right)

- Processing
  1. Generate n random numbers – called S
  2. (Store and Sort as Array)
     1. Store the numbers of S in an array: allocated with size n at the start
     2. **Sort using Bubble Sort**
  3. (Store and Sort as Vector)
     1. Store the numbers of S in a vector: allocated with size 1 and size increased one by one number until n
     2. Sort using Bubble Sort
  4. Measure time difference in both cases with n = <u>100</u>, <u>1,000</u>, <u>10,000</u>, <u>100,000</u>

- Output
  1. Make a graph showing any difference in timing
  2. Check the sorted results and confirm they are same (write a validator code to check)
  3. Document which one is your validation code

Algo 3 (BubbleSort):

- current_array = a = Input
- For (i=0; i<**N-1**; i++) {
  - For (**j=(i+1)**; j<N; j++) {
    - If(a[i] > a[j])
      - Swap(a[i], a[j])
  - }
- }
- **Return current_array**

Example invocation

> *SortN Bubble* 1000
Time for array: 1 sec
Time for vector: 2 sec
Result_checked: True

Home Work (#6) – C++ - Understand Code Optimization

- Consider BubbleSort algorithm for sorting (shown on right)

- Processing
  1. Generate n random numbers – called S
  2. (Store and Sort as Array)
     1. Store the numbers of S in an array: allocated with size n at the start
     2. **Sort using C++ Standard library's sort**
  3. (Store and Sort as Vector)
     1. Store the numbers of S in a vector: allocated with size 1 and size increased one by one number until n
     2. Sort using Bubble Sort
  4. Measure time difference in both cases with n = 100, 1,000, 10,000, 100,000

- Output
  1. Make a graph showing any difference in timing
  2. Check the sorted results and confirm they are same (write a validator code to check)
  3. Document which one is your validation code

See:
https://en.cppreference.com/w/cpp/algorithm/sort

Example invocation

> *SortN Standard* 1000
Time for array: 1 sec
Time for vector: 2 sec
Result_checked: True

Summary

- Program name: ***SortN***
  - Parameters: Bubble, Standard

- Test with: <u>100</u>, <u>1,000</u>, <u>10,000</u>, <u>100,000</u>

- Memory storage: Vector, Array

- Write a validator code to check result. Document which one is your validation code

- Draw a graph of results.

Peer Review: Homework Assignment #6

1. Go to spread sheet and on "Homework Assignments - Peer Review" tab. Go for today's date

2. Go to the row with your name

3. Peer review (10 mins)
    1. Enter <u>serial number</u> of person on your **LEFT** under "ID of code reviewer"
    2. Share code for the reviewer to see
    3. Reviewer: enter review (1-5)
    4. **Note**: negotiate – review code of neighbor or get own's code reviewed

4. Peer test (10 mins)
    1. Enter <u>serial number</u> of person on your **RIGHT** under "ID of code tester"
    2. Share command line for the tester to see
    3. Tester: enter review (1-5)
    4. **Note**: negotiate – test code of neighbor or get own's code tested

Peer Reviewing Guideline (10 mins)

- Look out for
  - Can you understand what the code is doing ?
  - Can you explain the code to someone else (non-coder) ?
  - Can you spot possible issues without running it?
    - Are the variables initialized ?
    - Are files closed?
    - Is their unnecessary code bloat ?

- What not to judge
  - Usage of language features, unless they are inappropriate

Assign rating

1: code not available
2: code with major issues
3: code with minor issues
4: -
5: no issues

Peer Testing Guideline (10 mins)

- Look out for
  - Does the program run as the coder wanted it to be (specification) ?
  - Does the program run as the instructor wanted it to be (requirement - customer) ?
  - Does the program terminate abruptly ?
  - Any special feature?

- What not to judge
  - Person writing the code

Assign rating

1: code not available
2: code runs with major issues
   (abnormal termination,
   incomplete features)
3: code runs with minor issues
4: -
5: No issues

Discussion on HW

- Peer Code Reviewing

- Peer Testing

Discussion: Course Project

Course Project – Knowing About Companies

- **Project**: Develop collaborative assistants (chatbots) that offer useful information about companies

- Specifically, use the EDGAR dataset on companies at: https://www.sec.gov/edgar/searchedgar/companysearch.
  - For Apple, it is: https://www.sec.gov/edgar/browse/?CIK=320193&owner=exclude

- **Each student will choose two companies (from thousand available).**

- Programming assignment programs will: (1) extract data about two companies from 10-k, (2) process it, (3) make content available in a command-line interface, (4) handle any user query and (5) report on interaction statistics.

Core Programs Needed for Project

- Prog 1: extract data from the district **[prog1-extractor]**

- Prog 2: process it (extracted data) based on questions **[prog2processor]**

- Prog 3: make content available in a command-line interface **[prog3-ui]**

- Prog 4: handle any user query **[prog4-userintent2querymapper]**

- Prog 5: report statistics on interaction of a session, across sessions **[prog5-sessionlogger]**

PA 6

- Project - the breakup of marks (100) will be as follows –
  - 20 points for the fully working demo. Submit code and video.
  - 40 points for report. Submit report in format.
  - 40 points for the presentation. Have slides ready.

- There will be no further submissions. **All are due by Monday, April 15, 2024, 10 pm**

Presentation for Code Reuse

For Extra Credits Only

Context -- PA6: Assembling the Final Chatbot *With Selective Code Reuse*

- One can possibly make more marks when doing final project assembly
  - **Remember**: PA1, PA2, PA3, PA4, PA5 will be the 5 programs from assignments. [100 points for each assignment]
  - **Remember**: Assembling code from one's on assignments gets the standard [100 points].
  - Extra points will be given if you make your code (for PA1 – PA5) available to others (make repository public) <u>AND</u> someone uses your code (any of PA1-PA5). Both will have to be reported in their individual project reports.
    - <u>**40 points**</u> **will be given per assignment to student whose assignment is reused**, and
    - <u>**20 points**</u> **will be given to person who reuses code**
  - Extra points will not exceed 100 points for any student. That is, one cannot make more than 700 points.

\<Student Name> - Project Summary

	Criteria	Information	What Needs to Change to Support Additional Company (Reuse Situation)
1	Companies and 10-k parts supported		
2	How data is extracted and stored		
3	How extracted data is processed, presented		
4	How intent is detected for user utterances		
5	How answer is given/ shown		
6	How statistics of session is calculated		
7	Performance of your chatbot	Quantitative, Qualitative	
8	Anything unique?		

Sample Output

Due One-Day Before Next Class

\<Student Name\> - Project Summary

Class Diagrams

Due Next Class

Prog 6: Assembling the Chatbot

- Have a program - **[my-chatbot]**

- User interacts with the chatbot with any utterance and the system has to answer – see right

- User can ask about statistics and query log
  - Same as PA5
  - See next slide

[#1] "Quit" or "quit" or just "q" => Program exits

[#2a]"Tell me about the CEO of IBM", "Tell me about the the risk factors"

[#2l]"Tell me everything" => Give all information Extracted (Type-I12)

[#3] "What companies do you support for Q/A" => Give list of companies supported

[#4] "Give me your usage stats" => Give chat summary info

[#5] *<User can enter any other text and the program has to handle it>* => "I do not know this information" or
   "Here is my guess - " + <query> + <answer>. "Did I answer correctly ? "

All Queries to be Supported

[#1] "Quit" or "quit" or just "q" => Program exits

[#2a]"Tell me about the CEO of IBM", "Tell me about the the risk factors"

[#2l]"Tell me everything" => Give all information Extracted (Type-I12)

[#3] "What companies do you support for Q/A" => Give list of companies supported

[#4] "Give me your usage stats" => Give chat summary info

[#5] *<User can enter any other text and the program has to handle it>* => "I do not know this information" or
   "Here is my guess - " + <query> + <answer>. "Did I answer correctly ? "

Chatbot usable in debug mode

my-chatbot –summary
=> There are 12 chats to date with user asking 23 times and system respond 24 times. Total duration is 456 seconds.
• **my-chatbot** –showchat-summary 2=> Chat 2 has user asking 2 times and system
respond 2 times. Total duration is 4 seconds.
• **my-chatbot** –showchat 2
=> Chat 2 chat is:
…
• **my-chatbot** –showchat 200
=> ERROR: there are only 12 chat sessions. Please choose a valid number.

PA6: Assembling the Final Chatbot *With Selective Code Reuse*

- One can possibly make more marks when doing final project assembly
  - **Remember**: PA1, PA2, PA3, PA4, PA5 will be the 5 programs from assignments. [100 points for each assignment]
  - **Remember**: Assembling code from one's on assignments gets the standard [100 points].
  - Extra points will be given if you make your code (for PA1 – PA5) available to others (make repository public) <u>AND</u> someone uses your code (any of PA1-PA5). Both will have to be reported in their individual project reports.
    - <u>**40 points**</u> **will be given per assignment to student whose assignment is reused**, and
    - <u>**20 points**</u> **will be given to person who reuses code**
  - Extra points will not exceed 100 points for any student. That is, one cannot make more than 700 points.
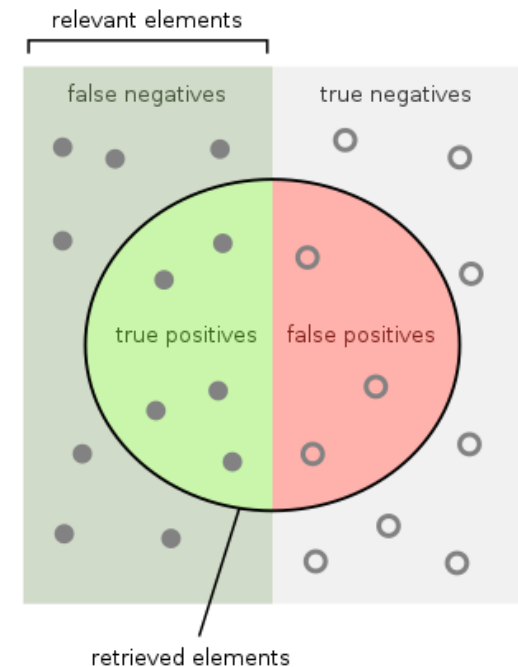
Project – PA#6

- Code organization
  - Create a folder in your GitHub called "**my-chatbot**"
  - Have sub-folders: src (or code), data, doc, test
  - Have data directory as shown in previous slide
    - ./data/**chat_sessions/**
    - ./data/ **chat_statistics.csv**
  - Write a
    - Report in ./doc sub-folder. Credit reuse
    - Create a presentation in ./doc sub-folder
  - Put a log of system interacting in ./test
  - Send a confirmation that code is done by updating Google sheet; optionally, send email to instructor and TA

- Use concepts learned in class
  - Exceptions
  - File operations
  - PA1 to PA5 from yourself or others; credit reuse in Readme, report and presentation

Evaluating a Chatbot's Performance Quantitatively (Information Retrieval)

- Precision:
  (# Correct answers / # answers retrieved) * 100
        // How well correctly answered

- Recall:
  (# Correct answers / # answers stored) * 100
        // How well correctly found

- Combined = F1 score =

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

relevant elements

false negatives   true negatives

true positives   false positives

retrieved elements

How many retrieved items are relevant?   How many relevant items are retrieved?

$$\text{Precision} = \frac{}{}$$ $$\text{Recall} = \frac{}{}$$

Format for Project Report –

- Requirement – What did the instructor ask you to do?

- Specification – What did you you do, what scope you selected and what decisions you made?

- Development highlights – How was your code implemented, e.g., module design, classes ? How did you test ? What problems did you face and how did you solve them?

- Reuse – What did you do to make your code reusable? Whose code did you use and why? Who is using your code and why ? What challenges did you face?

- Future work - What more can be done to make your chatbot useful? How will the code need to be changed over time?

Assignments: Late Submission Policy and Extra Marks

- There is no provision for late submission for programming assignments
  - Except when prior approval has been taken from instructor due to health reasons

- One can possibly make more marks when doing final project assembly
  - **Remember**: PA1, PA2, PA3, PA4, PA5 will be the 5 programs from assignments. [100 points for each assignment]
  - **Remember**: Assembling code from one's on assignments gets the standard [100 points].
  - Extra points will be given if you make your code (for PA1 – PA5) available to others (make repository public) AND someone uses your code (any of PA1-PA5). Both will have to be reported in project report.
    - **40 points will be given per assignment to student whose assignment is reused**, and
    - **20 points will be given to person who reuses code**
  - Extra points will not exceed 100 points for any student. That is, one cannot make more than 700 points.

Project Presenter Name:
Student Name:

Scope: Companies, Prog. Language

Data: What data is available and what is retrieved from program ?

Code Organization: Anything significant
               to highlight ?
 PA1:
PA2:
…
 PA6: code reuse by someone, and of someone

Queries Snapshot

Video link:

Experience implementing the chatbot, Testing

Experience with reuse

Review of Main Concepts

Review of Topics

Class #	Date	Description	Comments
1	Jan 9 (Tu)	Introduction	
2	Jan 11 (Th)	Introduction – Pointers, Iteration	
3	Jan 16 (Tu)	Input/ Output	
4	Jan 18 (Th)	I/O, Exceptions	HW 1 due
5	Jan 23 (Tu)	Memory management, User defined types	Prog 1 - start
6	Jan 25 (Th)	Object Oriented (OO) intro	HW 2 due
7	Jan 30 (Tu)	OO concepts, UML Notations	
8	Feb 1 (Th)	Code org (C++)	Prog 1 - end
9	Feb 6 (Tu)	OO – inheritance	Prog 2 - start
10	Feb 8 (Th)	Regex, OO - polymorphism	HW 3 due
11	Feb 13 (Tu)	Exceptions	
12	Feb 15 (Th)	OO – Constructor, Destructor	Prog 2 – end
13	Feb 20 (Tu)	Review: inheritance, Polymorphism	Prog 3 - start
14	Feb 22 (Th)	In class test	Quiz 1 – In class
15	Feb 27 (Tu)	In class Project Review: PA1 and PA2	
16	Feb 29 (Th)	OO – operators, access control	Prog 3 - end Semester - Midpoint
	Mar 5 (Tu)		Spring break – No class
	Mar 7 (Th)		Spring break – No class

Class #	Date	Description	Comments
17	Mar 12 (Tu)	C++ standard library, Testing strategies	Prog 4 - start
18	Mar 14 (Th)	Advanced: Pointers	HW 4 due
19	Mar 19 (Tu)	Advanced: Pointers, I/O	
20	Mar 21 (Th)	Advanced: Operator overloading	Prog 4 – end
21	Mar 26 (Tu)	Advanced: Memory Management	Prog 5 – start HW 5 due
22	Mar 28 (Th)	Advanced: Code efficiency	
23	Apr 2 (Tu)	Advanced: Templates	
24	Apr 4 (Th)	AI / ML and Programming	Prog 5 – end
25	Apr 9 (Tu)	Project code summary – student presentation for reuse Review material for Quiz 2	HW 6 due Prog 6 – assembling start
26	Apr 11 (Th)	In class test	Quiz 2 – In class
27	Apr 16 (Tu)	Project presentation	Prog 6 - due
28	Apr 18 (Th)	Project presentation	Last day of class (April 22 per bulletin)
	Apr 23 (Tu)		Reading Day
29	Apr 25 (Tu)	9am – Exam or Final Overview	Examination

Concluding Section

Lecture 25: Concluding Comments

- Quiz 6 peer evaluation

- PA6 and reuse presentation

- Review for Quiz 2

About Next Lecture – Lecture 26

Lecture 26: Quiz 2

- Quiz 2 – depends on reuse for PA 6
  - All concepts taught in class
  - No online test giving option

19	Mar 19 (Tu)	Advanced: Pointers, I/O	
20	Mar 21 (Th)	Advanced: Operator overloading	Prog 4 – end
21	Mar 26 (Tu)	Advanced: Memory Management	Prog 5 – start HW 5 due
22	Mar 28 (Th)	Advanced: Code efficiency	
23	Apr 2 (Tu)	Advanced: Templates	
24	Apr 4 (Th)	AI / ML and Programming	Prog 5 – end
25	Apr 9 (Tu)	Project code summary – student presentation for reuse Review material for Quiz 2	HW 6 due Prog 6 – assembling start
26	Apr 11 (Th)	In class test	Quiz 2 – In class
27	Apr 16 (Tu)	Project presentation	Prog 6 - due
28	Apr 18 (Th)	Project presentation	Last day of class (April 22 per bulletin)
	Apr 23 (Tu)		Reading Day
29	Apr 25 (Tu)	9am – Final Overview	Examination