AI INSTITUTE #AIISC
UNIVERSITY OF SOUTH CAROLINA

UNIVERSITY OF SOUTH CAROLINA

CSCE 240: Advanced Programming Techniques
Lecture 13: Exceptions, Error Handling

PROF. BIPLAV SRIVASTAVA, AI INSTITUTE

22ND FEBRUARY 2022

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.

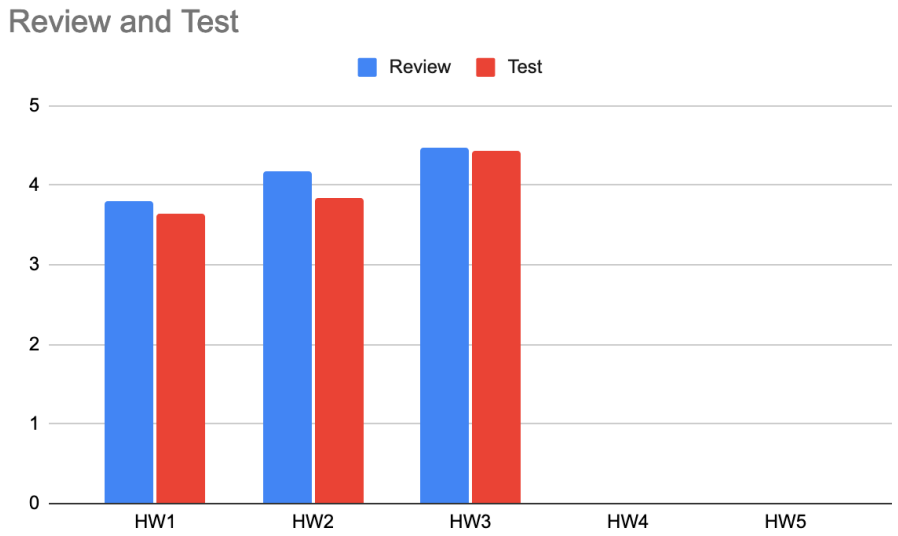
Organization of Lecture 13

- Introduction Section
  - Recap of Lecture 12
  - TA and SI Updates
- Main Section
  - Concept: Errors
  - Concept: Exceptions, for error handling
  - Discussion: Project
- Concluding Section
  - About next lecture – Lecture 14
  - Ask me anything

Introduction Section

Recap of Lecture 12

- Review of Quiz 1
- Peer review of HW3
  - Improving quality of classes' code
- Review of Inheritance
  - Concept: Inheritance Type
- Review of Polymorphism



Announcements

- Chatbots – Event on March 18, 2022
  - Collaborative Assistants for Society (CASY) – in person and virtual event on campus
  - 9:30 am – 1:00 pm; talks and student use-cases

- Details and registration info: https://casy.aiisc.ai

Updates from TA, SU

- TA update: Yuxiang Sun (Cherry)
- SI update: Blake Seekings

Main Section

Concept: Errors

What is an Error ?

- **Error**: Anything that is not as-expected
- Errors at different levels
  - **Conceptual**: at the problem and solution approach level
  - **Implementation**: in the program
  - **Ongoing / runtime**: while running

Types*

[interface error, logic error]

[syntax error, compilation error, arithmetic error]

[resource error, runtime error]

*Credit: https://textexpander.com/blog/the-7-most-common-types-of-errors-in-programming-and-how-to-avoid-them

Why There are Errors ?

- Conceptual: at the problem and solution approach level
  - Customer did not make the requirement clear (requirement)
  - Developer did not understand the problem clearly (specification)
- Implementation: in the program
  - Programming concepts were used wrongly, poor coding
  - Test cases were exhaustive
- Ongoing / runtime: while running
  - World changed, and so did problem, solution
  - Runtime environment – resources or data, changed

*Credit: https://textexpander.com/blog/the-7-most-common-types-of-errors-in-programming-and-how-to-avoid-them

Error Handling

- Objective
  - Program has predictable behavior
    - Usually, terminate with a message
    - Optional: tries to recover
  - Developer gets hints to improve the code

- Typical pattern of error handling by a developer

```
check_condition

if (abnormal) {
    // print message
    // terminate
}
```

Error Handling via Exception Mechanism

- Most languages have an exception mechanism to ***anticipate*** abnormal situations and do something about those ***rare*** cases

- Typical pattern of using exceptions in programming language

```
try {
        // developer anticipates

} catch {
        // do something about abnormal situation

        // print message
        // terminate
}
```

Exception in C++

- Demonstration
  - Using exception for string out-of-range
  - Custom exception handling
- Discussion
  - Possible to have multiple handlers
  - Can throw exception too

Exception Handling in Java

- Demonstration
  - Using exception for string out-of-range

- Discussion

Exception Handling in Python

- Demonstration
  - Using exception for string out-of-range

- Discussion
  - Multiple exception handlers
  - Specialized handler called if specified

Common Use-Cases for Exception Handling

- Input/ Output
  - Files, Streams not found
  - Runtime errors
- String manipulation
- Arithmetic errors – e.g., divide by zero

Discussion: Course Project

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.

Core Programs Needed for Project

- Prog 1: extract data from the district
- **Prog 2: process it (extracted data) based on questions**
- Prog 3: make content available in a command-line interface
- Prog 4: handle any user query and
- Prog 5: report statistics on interaction of a session, across session

Programming Assignment # 2

- Goal: **process extracted text based on questions**
  - Language of choice: Any from the three (C++, Java, Python)
- Program should do the following:
  - Take input from a local file with whose content is obtained from Prog#1 (when district name given as input)
  - Given an information type as input, the program will return its content
    - Examples: Contact Information, personal information, voting records
    - Input type can be given as command line argument.
      Examples:
      - prog2processor –t “Contact Information”
      - prog2processor –t “Contact Information:name” // Get person’s name
  - For demonstrating that your program works, have a file called “test_output.txt” showing the set of supported commandline options and output in the doc folder.
- Code organization
  - Create a folder in your GitHub called “prog2-processor”
  - Have sub-folders: src (or code), data, doc, test
  - Write a 1-page report in ./doc sub-folder
  - Send a confirmation that code is done to instructor and TA, and update Google sheet

- Contact Information (Type-I1)
  - Name
  - Region
  - Addresses: Columbia, Home
  - Phone: Business, Home
- Personal Information (Type-I2)
- Committee Assignments (Type-I3)
- Sponsored Bills in the House (Type-I4)
- Voting Record (Type-I5)
- **Service in Public Office (Type-I6)**

Example: Representative Information

Input:

prog2processor –t “Contact Information:name” // Get person’s name

Output:

Terry Alexander

- Contact Information (Type-I1)
- Personal Information (Type-I2)
- Committee Assignments (Type-I3)
- Sponsored Bills in the House (Type-I4)
- Voting Record (Type-I5)
- Service in Public Office (Type-I6)

Representative Terry Alexander

Democrat - Florence
District 59 - Darlington & Florence Counties - **Map**

Columbia Address
314C Blatt Bldg.
Columbia 29201

Home Address
1646 Harris Court
Florence 29501

Business Phone (803) 734-3004

Home Phone (843) 665-7321

Send message to Representative Alexander

Personal Information

- Education Consultant & Pastor
- Residing at 1646 Harris Court, Florence
- Born January 23, 1955 in Florence
- Son of the late James and Adell Alexander
- Durham Business College, A.D., 1976
- Francis Marion University, B.A., 1991
- Howard University School of Divinity, M. Div., 1998
- Married to Starlee Davis Alexander, 2 children, Terrell McClain and Matthew
- Pastor, Wayside Chapel Baptist Church
- Career Development Consultant
- Adjunct Professor of Religion, Limestone College
- Pee Dee Regional Council of Governments
- Past President, Habitat for Humanity, Board of Directors
- Charter member, The Florence Breakfast Rotary Club
- Past President, Boys and Girls Club of Florence
- Boy Scouts of the Pee Dee Executive Boards
- Florence Branch, NAACP, past President
- Mercy Medicine Board
- Pee Dee Chapter American Red Cross
- 100 Black Men of the Pee Dee
- Kappa Alpha Psi Fraternity, Inc.
- Francis Marion Society
- National Association of County Officials
- National Association of Black County Officials
- South Carolina Association of Black County Officials
- South Carolina Association of Guidance Counselors
- South Carolina Alliance of Black Educators

Committee Assignments

- **Education and Public Works, *2nd V.C.***
- **Regulations and Admin. Procedures**

Sponsored Bills in the House

- Primary Sponsor: (•) Yes ( ) No
- Search Session: 2021-2022 (124) Find Bills

Voting Record

- Search Session: 2021-2022 (124) Find Votes

Service In Public Office

- Florence County Council, 1990-06, District Number 3
- House of Representatives, 2007 - Present

Reminder: 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

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.

Concluding Section

Lecture 13: Concluding Comments

- We looked at the concept of exception
  - Errors are inevitable, handling has to be in place
  - Exception provides developer a way control behavior when rare situations occur; usually runtime
- Programming Assignment #2 is due

About Next Lecture – Lecture 14

Lecture 14: Constructors / Destructors

- We will discuss constructors and destructors in detail
- Launch of programming assignment #3
- Home work #4 will be given

14	**Feb 24 (Th)**	**OO – Constructor, Destructor**	**Prog 3 - start**
15	Mar 1 (Tu)	OO – operators, access control	HW 4 due
16	Mar 3 (Th)	C++ standard library	Prog 3 - end Semester - Midpoint