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

PROF. BIPLAV SRIVASTAVA, AI INSTITUTE

21ST 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
  - Announcements
  - Recap of Lecture 12
- Main Section
  - Concept: Errors
  - Concept: Exceptions, for error handling
  - Discussion: Project
- Concluding Section
  - About next lecture – Lecture 14
  - Ask me anything

Introduction Section

Announcements

- Programming Assignment #1: marks posted
- Quiz 1: marks posted

PA: Code Reviewing Rubric Used

- Look out for
  - Can one understand what the code is doing ?
  - Can one explain the code to someone else (non-coder) ?
  - Can one 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

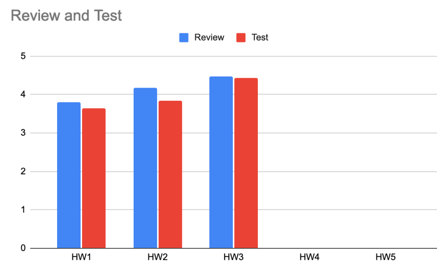
Assign rating (out of 100 -/+)
- -100: code not available
- -80: code with major issues
- -60: code with minor issues
- -20:
- 0: (full marks): no issues
- +20: special features

PA: Code Testing Rubric Used

- 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 ?
  - Is there a hardcoding of directory ? Paths should be relative to code base directory.
  - Any special feature?
- What not to judge
  - Length of documentation. It can just be short and accurate.
  - Person writing the code

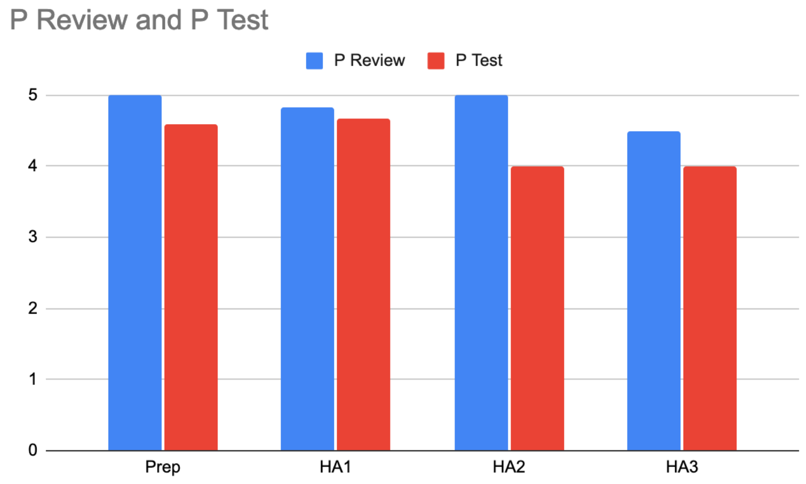
Assign rating (out of 100 -/+)
- -100: code not available
- -80: code with major issues (e.g., abnormal termination, incomplete features)
- -60: code with minor issues
- -20:
- (full marks): no issues
- +20: special features

Reference:
Spring 2022
(45+ students)



Recap of Lecture 12

- Review of Quiz 1
- Peer review of HW3
  - Slight fall in quality of code OR better peer testing
  - Caveat: Sample size is small
- Review of Inheritance
  - Concept: Inheritance Type
- Review of Polymorphism



Spring 2023

Announcements

- Quiz 1 has been graded
-

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

Level	**Types***
Conceptual	[interface error, logic error]
Implementation	[syntax error, compilation error, arithmetic error]
Ongoing / runtime	[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
  - Poor coding
  - Programming concepts were used wrongly
  - Test cases were exhaustive
- Ongoing / runtime: while running
  - World changed, and so did problem, solution
  - Runtime environment – resources or data, changed

Credit: Anonymous Creator

while (not edge) {
run();
}
do {
run();
} while (not edge);

Difference between "while" and "do-while"

*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

- Example 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
}
```

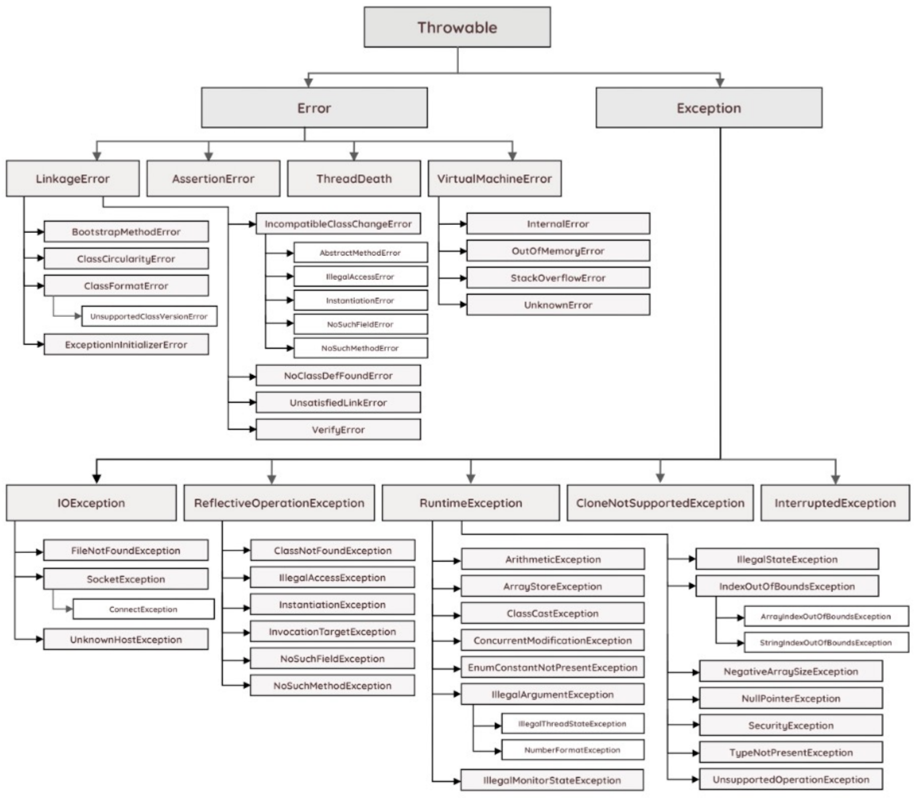
Exception in C++

- 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
  - All exceptions have a super-class, Exception



Credit: https://rollbar.com/blog/java-exceptions-hierarchy-explained/#

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 – 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.

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 **disease** name given as input)
  - Given an information type as input, the program will return its content
    - Examples: what is disease (I1), who is at risk (I2), disease vaccine (I12)
    - Input type can be given as command line argument. Examples:
      - prog2processor –t "what is **malaria**?" // Tell about disease
      - prog2processor –t "more information" // Get more info
  - 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

S1: https://www.cdc.gov/travel/diseases/malaria

- What is malaria? [I1]
- Who is at risk? [I2]
- What can travelers do to prevent malaria? [I3]
- After Travel [I4]
- More Information [I5]

S2: https://www.webmd.com/a-to-z-guides/malaria-symptoms

- What Is Malaria? [I1]
- Malaria Causes and Risk Factors [I2]
- Types of Malaria [I6]
- Symptoms [I7]
- When to Call a Doctor About Malaria [I8]
- Malaria Diagnosis [I9]
- Malaria Treatment [I10]
- Malaria Complications [I11]
- Malaria Vaccine [I12]

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

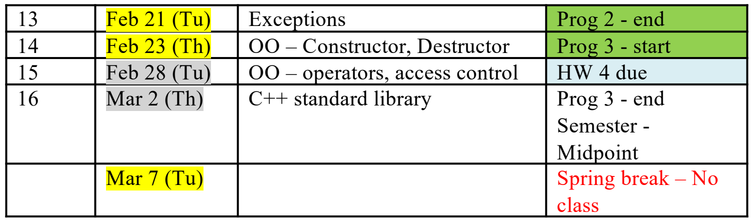
Lecture 13: Concluding Comments

- 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 by 10pm

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



13	Feb 21 (Tu)	Exceptions	Prog 2 - end
14	Feb 23 (Th)	OO – Constructor, Destructor	Prog 3 - start
15	Feb 28 (Tu)	OO – operators, access control	HW 4 due
16	Mar 2 (Th)	C++ standard library	Prog 3 - end Semester - Midpoint
	Mar 7 (Tu)		Spring break – No class