

Exceptions

CSSE 221

Fundamentals of Software Development Honors

Rose-Hulman Institute of Technology

Announcements

- Pass in quiz from last class and the runtime assignment graphs/explanations now to your in-class TA
- The LUG has its install fest on Wednesday 09/18 in O-169 from 6-9 PM.
- Questions?

- Other questions?

Answers to runtime assignment

1. A single for loop from 1 to n gives code that is $O(n)$.
2. Sequential for loops, each running from 1 to n , are still $O(n)$
3. Nested for loops, each running from 1 to n , are $O(n^2)$
4. Nested for loops in which $f()$ is called once from within the inner loop and once outside the inner loop, but still inside the outer loop, are still $O(n^2)$.

When processing an array of size n ...

1. ...if you just have to look at each entry individually: $O(n)$
2. ...if you have to look at each pair of entries individually: $O(n^2)$

- If you got any of these wrong, ask me or an assistant before the end of the week (so you can learn and reinforce the right understanding before you complete HW2)

Capsules Phase 1: research and summarize

1. Read
2. Extract important concepts
3. Write a **single (or no more than two) page summary**
4. Write a short quiz (5 to 8 of your own questions)
5. Include a key
6. Proofread!

Good quizzes...

1. Aren't too long or too short (5-8 of your questions, ~5 mins total)
2. Contain questions that vary in difficulty
3. Are consistent with the summary
4. Contain questions that are clearly worded and unambiguous
5. Include an answer key with correct answers

Notes

- **Due Date:** email the quiz, key, and summary to me 48 hours before the class in which they will be used
- A summary is posted for today
- A grading rubric is posted on the schedule
 - It was emailed to you

This week: BigRational assignment

- Tuesday:
 - API (Application Programming Interface)
 - Interfaces: writing to a contract
- Wednesday:
 - Unit Testing: searching for logic errors
 - Introduction to efficiency analysis: “big-Oh”
- Today:
 - Exceptions: throwing and catching

Exceptions

- Used to handle *unusual* situations.
- Vocabulary: *throw, try, catch, finally, throws*
 - Football or hot-potato analogy
 - A piece of code will **try** to perform an operation. If something unusual happens, then the code will **throw** an exception. It is propagated through the methods that called it, via **throws**, until it is **caught**. **Finally**, some code will execute whether an exception was thrown or not.

Onto the assignment

- Click the “Throwing Exceptions” link on the schedule page.
- When you finish, you may work on BigRational.