Welcome to ComS 228 Recitation

Andrew Kicklighter

- Third semester as a TA for ComS 228
- Junior at Iowa State in Computer Engineering
- Fun fact about me: I've spent my last 3 summers (and this coming summer) in New Mexico at Philmont Scout Ranch being paid to backpack in the backcountry. In that time span I've climbed 6 14ers (14,000 ft or higher peaks)
- Email: apkick@iastate.edu
- Office Hours: Immediately after this recitation (Thursday 11:00-11:50)

Katelyn Lamison

- Sophomore in Computer Science
- First semester TAing for Com S 228
- Office Hours Wed. 3:10-4pm
- klamison@iastate.edu

Why is this course important?

- Concepts are used in future courses
- Technical Interviews
- Internships/Jobs
- You need to pass this class to graduate

What's the point of recitation if it's not required?

- A more in depth look at material covered in class
- A place to ask TAs questions
- Answer questions about assignments
- Review for exams
- Hand exams back

How to succeed in ComS 228

- Attend office hours **EARLY**
- Start projects when they come out, not the day before they’re due
- Ask questions!
- Come to recitation

Today’s Concepts

- Command Line Arguments
- Debugging
- JUnit

Command Line Arguments

- Will be used in upcoming programming assignments.
- How To:
  - Right Click (Ctrl click on Mac) on java class you want to run
  - Hover Over “Run As”
  - Click on “Run Configuration”
  - Click “Arguments” Tab
  - Write arguments within “Program Arguments” text box
    - They will be saved as a String Array
    - Input: Hello World 1
    - Output: [“Hello”, “World”, “1”]
  - Click “Apply” to apply changes

Debugging

- Why Debug?
  - To find where in your code the errors exist.
- What can you do with debugging?
  - Perspectives
  - 'Play' button vs. "Bug" button: (Compiles and runs vs. attaches debugger as well)
  - Setting breakpoints
  - Regular and conditional
  - Step in/over/return at breakpoint
  - Inspecting variables
  - Watching variables
  - Editing variables
  - Filtering Packages to Step into
- Examples: https://youtu.be/dLeZ-oZZw7U

JUnit Tests

- Why Test?
  - Code can be tricky
  - Many developers on one project can cause errors
  - Want to pass/get an A on the programming assignments
- When to test?
  - Whenever you write code!
- How to Test in Java:
  - Use JUnit!
  - NOT MAIN!!
  -

How to create a JUnit Test Case

1. Right Click (Ctrl click on Mac) on package you want to create JUnit Test Class
2. Hover Over “New”
3. Click on “Other”
4. Click on “Java” Dropdown
5. Click on “JUnit” Dropdown
6. Click on “JUnit Test Case”
7. Click “Next” at bottom of dialog
8. Enter in Test Class Details (Name, Superclass, etc)
9. Click “Finish” at bottom of dialog
10. Pop up might come up asking to add JUnit 5, click yes

JUnit Annotations

- @BeforeClass and @AfterClass
  - Think them as one time “setup” and “tearDown”. They run for one time before and after all tests
- @Before and @After
  - Annotations for “setup” and “tearDown” respectively. They run before and after every test case
- @Test
  - Your class does not need to extend from “TestCase” class
- @Test(expected = ExceptionType.class)
  - Use “expected” parameter with @Test annotation for test cases that expect exception.

Useful Asserts

- assertTrue(Condition) & assertFalse(Condition)
  - It ensures the condition is true or false
- assertNull(Variable) & assertNotNull(Variable)
  - This assert ensures a variable is null or it is not null.
- assertEquals(Expected, Actual)
  - It ensures equality between actual and expected.

JUnit Example

How to access these files?

- You can find it on our Recitation Page.
- How do you access our Recitation Page?
  - Go to the Canvas Com-S-228 course page
  - On left side bar, click on modules
  - Our Section will be under
    - Recitation Material -> Andrew and Katelyn
- What will you find in our Recitation Page?
  - Code we went over today
  - These Slides

Questions?