Testing a class; Control Structures

- From last time: Shadowing
- Test programs: **`StudentTester`**
  - incremental development
  - enhancing the **`Student`** class
- Control structures
  - **`if`** statements
    - Ex: **`PizzaCalc.java`**
  - Error-checking input
    - Ex: **`NapCalc.java`**
  - Multi-way tests
    - Ex: **`getGrade`**
  - Other topics from textbook

Announcements

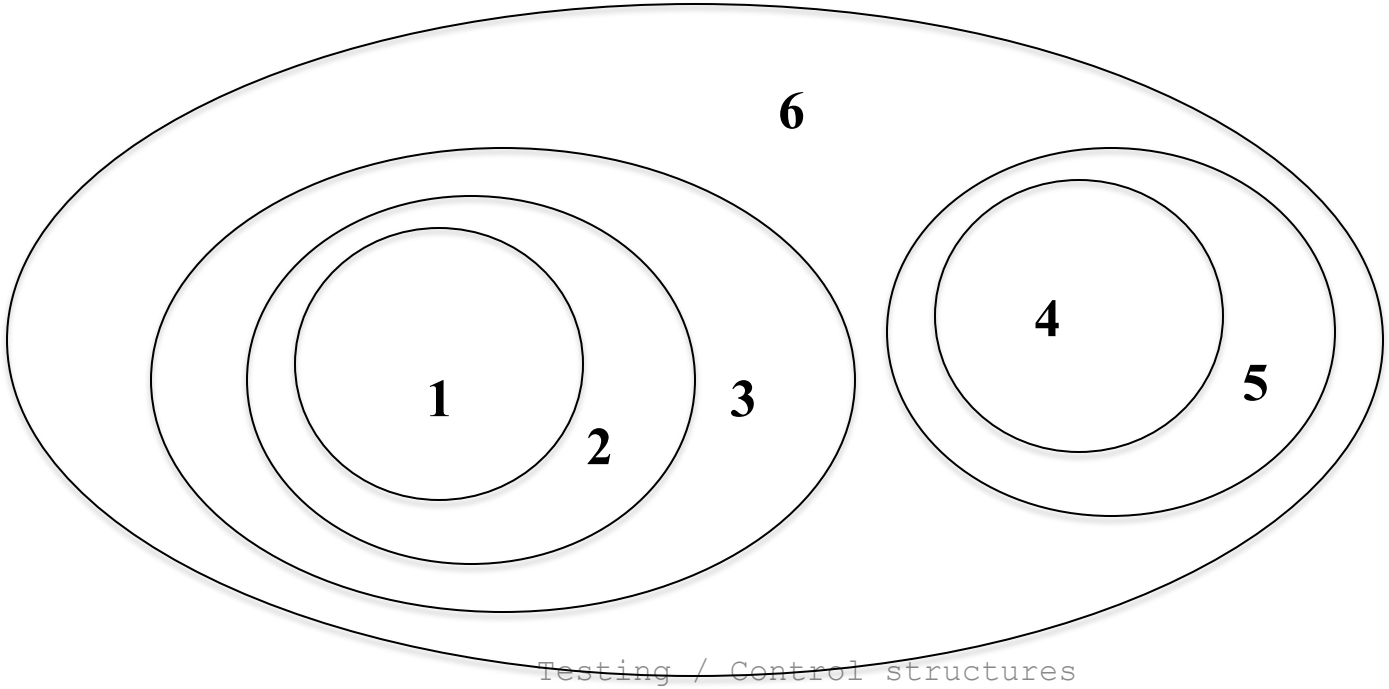
- Reminders:
  - PA1 due in a little more than a week
  - Lab 3 has advanced prep: see lab for details
- Students that need to see me after class or in office hours today:
  - missed the first lecture
  - not officially enrolled
  - who have no previous programming experience

Test programs

- **`Student`** class is not a complete program
- Could use in some larger app, or…
- first create a class to test the **`Student`** class:
  - **`StudentTester`**, will contain **`main`** method.
  - each class in it's own file:
    this one is in **`StudentTester.java`**

Incremental development

- App with 20 classes
- BAD: type in all the code and compile and run
- BETTER: test each class independently and make sure it works before integrating it:



Developing code for one class

- Within-class incremental development
  - Add functionality to a class incrementally and test as you go.
  - Minimal: constructor plus accessors
- Test-driven design
  - design test cases and compute expected output BEFORE writing any code
- Let's enhance **`Student`** class, and use **`StudentTester`** to test the changes.

`if` statement example

- let's compile and run **`PizzaCalc.java`**

Control Structures

block indentation conventions

- always use curly brackets:

```
if (cond) {
   action;
}
```

```
if (cond)
{
    action;
}
```

- line up right curly bracket with start of `if`
- body of `if` indented 3 spaces further
- left curly on same line or following line (both shown above)

Error-checking input

- Do **`NapCalc.java`** example.

Multi-way tests

- mutually exclusive conditions:
  - if-else-if construct
  - Don't keep nesting (and indenting):

```
if (cond) {
    action1;
}
else if (cond2) {
   action2;
}
else {
   action3;
}
```

Multi-way test example

- Ex: assign letter grade based on score in course: 90, 80, 70, 60

```
public static char getGrade(int score)
```

Multi-way test example (cont.)

- Goes with POLL
- Ex: assign letter grade based on score in course:
  90, 80, 70, 60 are cutoffs.
- Consider a new version of the code:

```
public static char getGrade(int score){
  char grade = 'F';
  if (score >= 90) { grade = 'A'; }
  if (score >= 80) { grade = 'B'; }
  if (score >= 70) { grade = 'C'; }
  if (score >= 60) { grade = 'D'; }
  return grade;
}
```

Asynchronous participation: Link to Multi-way Test poll

Multi-way test 2nd example

- Interpret one-word commands for an interactive console-based program.

  `CommandProcessor.java`

- Do this example on your own later (start with code that goes with today's lecture)

Other important control structures topics in the textbook

- The dangling-else problem: Common Error 5.3
- DeMorgan's laws: Special Topic 5.7
- Hand-tracing: Prog. tip 5.5 and Section 6.2
- Another example of multi-way test: Do **`CommandProcessor.java`** example (starter code in 02-02 Lecture Code on vocareum)

(don't just read, but practice too)