RADFORD UNIVERSITY

ITEC 120

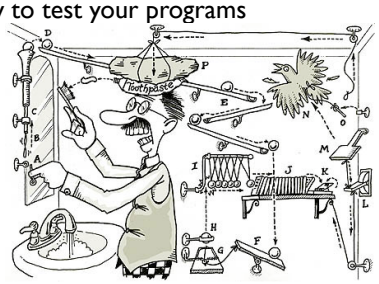
Lecture 11
Testing

Review

- Questions?
- Designing programs

Objectives

- How to test your programs



Motivation

- Your program prints out 103945
- It is supposed to print out 1024
- It contains 6 functions that mix stair stepping and conditional execution
- How would you fix the bug?

Testing

- Think of 3 different ways to tell if your code works?
- Come up with one advantage and disadvantage for each approach.

Why?

- If you don't test your code:

Print debugging

Peer into what your program is doing.

It may surprise you!

```
int input=scan.nextInt();
System.out.println(input);
String mystery = scan.nextLine();
System.out.println(mystery)

if (mystery.length() > input)
{
    System.out.println("In if block");
    this.otherFunc(mystery);
}
else
{
    System.out.println("In else block");
    this.anotherFunc(input);
}
```


Tracing

- Write down each function
- Write down variables for each function
- Write down values of variables as they change

Brute force

- Does it work on this case?
- Fix code until yes.
- Repeat step 1 until exhausted or deadline

Gradual

- Small, medium and large tests
  - Read input, print out input
  - Does it execute one command correctly
  - Does it execute two-three commands correctly

Unit testing

- Instead of compile / run, come up with own tests. Compile, run against test suites
- Generate tests before writing code

```
public void testFunctions()
{
   System.out.println("Expected distance = 1");
   System.out.println(this.calcDistance(0,0,0,1,0,0));
   System.out.println("Expected number=4");
   System.out.println(this.square(2));
}
```


Clean room

- Mathematical proof that code functions in a particular manner
- Advantage?
- Disadvantage?
- Where do you think it is used?

Question

- Should you write the program first after reading the problem or should you write the tests?
- Take a minute to decide and come up w/ 2 supporting arguments.

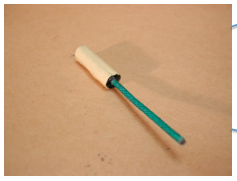
Testing

Test driven development

- Understand problem
- Decompose into parts
- Design tests for each part and interactions
- Implement part, test it
- Integrate with other parts, test

Testing

Edge cases



Long fuse

Short fuse

Testing

Edge cases

- When someone makes more than 50,000 a year, raise their tax rate to .35
- Test before (<50,000)
- Test at (50,000)
- Test after (>50,000)

Can be a Bigger deal than you think

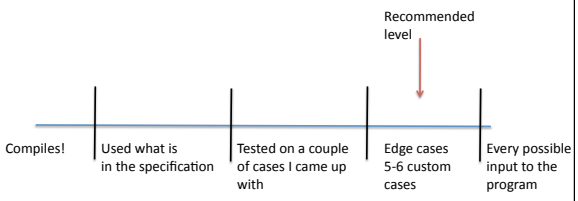
Testing

Code

```
int fuseLength = fc.getLength();
if (fuseLength > 15)
{
      fc.fire();
}
else if (fuseLength == 15)
{
      System.out.println("You are at the minimum fuse length, continue?");
      String answer = Scan.nextLine();
      if (answer.equals("yes"))
            fc.fire();
}
else
{
      System.out.println("Error, fuse not long enough to ignite safely");
}
```

Testing

How much?



Testing

Summary

- Testing
  - Rationale
  - Example / Industry methods
  - Edge cases
  - How much

Testing