

Web-CAT and PA3 Hints




Dr. Chris Mayfield

Department of Computer Science
James Madison University

March 03, 2014



Web-CAT scores

Score Summary		
Design/Readability:	<input type="text"/> /10.0	«Awaiting Staff»
Style/Coding:	20.0/20.0	
Correctness/Testing:	70.0/70.0	
Final score:	90.0/100.0	

(10 pts) Design/Readability

- ▶ Graded manually – comments, methods, variables, etc.

(20 pts) Style/Coding

- ▶ Graded by Checkstyle – 1 or 2 points off per violation

(70 pts) Correctness/Testing

- ▶ Your own tests – should be 100% before submitting
- ▶ Code coverage – how much of your code did you test?
- ▶ Problem coverage – how many of my test cases passed?

Web-CAT results

Click the filename for **code coverage** hints

- ▶ Mouse over red lines for details

```
case CHANCE:
```

```
    score = dice.addValue();
```

```
    break;
```

```
default:
```

This line was never executed by your tests.

Be sure to read the **problem coverage** hints

▾ Estimate of Problem Coverage (33%)

Problem coverage: 33%

For this assignment, the proportion of the problem that is covered by your test cases is being assessed by running a suite of reference tests against your solution, and comparing the results of the reference tests against the results produced by your tests.

Differences in test results indicate that your code still contains bugs. Your code appears to cover **only 33%** of the behavior required for this assignment.

Your test cases are not detecting these defects, so your testing is incomplete—covering at most **only 33%** of the required behavior, possibly even less.

Double check that you have carefully followed all initial conditions requested in the assignment in setting up your solution, and that you have also met all requirements for a complete solution in the final state of your program.

The following hints may help you locate some ways in which your solution and your testing may be improved:

- Test your program using all zeros for input.
- Invalid report format on line 3

Example feedback:

*Test your program using
all zeros for input.*

*Invalid report format on
line 3.*

PA3 coverage hint

Your test cases must **cover every statement**

```
/**
 * Test default constructor and invalid category.
 */
public void testDefaults() {
    new Yahtzee();
    assertEquals(0, Yahtzee.calculateScore(-1, null));
}
```

- ▶ The first line tests the class definition
- ▶ The second line tests the **default** case

DrJava auto-indent bug

Put if under case, press Ctrl-A + Tab

```
case FULL_HOUSE:
    if (isFullHouse(dice)) {
        score = 25; // should be indented
    }               // should be indented
    break;
```

A better way to define your methods

```
case FULL_HOUSE:
    score = fullHouse(dice); // returns 0 or 25
    break;
```

Indenting if statements

NO

```
if(category == LARGE_STRAIGHT) {  
    score = 0;  
}
```

OK

```
if (category == LARGE_STRAIGHT) {  
    score = 0;  
}
```

Notes

- ▶ `if` is not a method
 - ▶ Don't do: `System.out.println ("extra space");`
- ▶ Always use braces with `if`
 - ▶ Even if body is just one line!

Indenting switch statements

NO

```
switch (category) {  
    case FULL_HOUSE: System.out.println("Full house");  
        break;  
    default: System.out.println("Invalid");  
        break;  
}
```

OK

```
switch (category) {  
    case FULL_HOUSE:  
        System.out.println("Full house");  
        break;  
    default:  
        System.out.println("Invalid");  
        break;  
}
```

Counting dice values

From now on, it's okay to combine **declaration** and **initialization**

```
public static int fullHouse(Dice dice) {  
    int c1 = dice.countValues(1);  
    int c2 = dice.countValues(2);  
    int c3 = dice.countValues(3);  
    int c4 = dice.countValues(4);  
    int c5 = dice.countValues(5);  
    int c6 = dice.countValues(6);  
    int score = 0;  
    // figure out if it's a full house  
    return score;  
}
```