**CS 220 – Data Abstraction**

**HW 1 – Testing and Debugging**

**HW 1 Due: 2200, Lesson 5, Tuesday, 17 January**

|  |
| --- |
| Help Policy: **AUTHORIZED RESOURCES:** Any, except another cadet’s program.  **NOTE:**   * Never copy another person’s work and submit it as your own. * Do not jointly create a program unless explicitly allowed. * You must document all help received from sources other than your instructor. * **DFCS will recommend a course grade of F for any cadet who egregiously violates this Help Policy or contributes to a violation by others.**  Documentation Policy:  * You must document all help received from any source other than your instructor. * The documentation statement must explicitly describe WHAT assistance was provided, WHERE on the assignment the assistance was provided, and WHO provided the assistance. * If no help was received on this assignment, the documentation statement must state “NONE.” * If you checked answers with anyone, you must document with whom on which problems. You must document whether or not you made any changes, and if you did make changes you must document the problems you changed and the reasons why.  Vague documentation statements must be corrected before the assignment will be graded, and will result in a 5% deduction on the assignment.Turn-in Policies:  * On-time turn-in is at 2200 on the due date, same day for both M and T day sections. * Late turn-ins will receive a 25% penalty per 24 hours late unless prior arrangements have been made with your instructor. * There is no early turn-in bonus or extra credit for this assignment. |

1. Objectives

* Be able to read, understand, and modify a significant amount of code written by another programmer
* Be able to use JUnit test capabilities to find and fix errors in code

1. Assignment

There are errors in the provided shape pad code. You must create JUnit test cases for all the shapes, locate, and fix the errors. The errors present are non-obvious. Merely running the program and testing it may not find the errors. More extensive unit testing must be done. You should focus your efforts on testing the area and contains method for each shape. There are six shapes to test, which means you should have twelve JUnit test routines. Each routine should contain multiple test cases to find the errors. Specifically, for each shape and each method, you should test

* values that are obviously correct,
* values that are obviously incorrect, and
* boundary values.

Your tests should help you zero in on the errors if they are defined carefully.

Once you find the errors using your JUnit tests, fix them. Include an obvious commenting statement in the code that describes what the error was and how you fixed it.

1. Homework Submission Requirements

* Fill in your name, the documentation statement, and self-assessment in the pex grade sheet file in the top level of the downloaded NetBeans project folder!
* Your NetBeans project name must be LastnameHW1.
* Zip this entire folder to LastnameHW1.zip and submit it using the Webpost link on the left side of the course web page.
* Submit only one file for the programming exercise, a zip file containing your entire NetBeans project with the PEX Grade Sheet document in the top-level folder.

**CS 220 – HW 1 – Grade Sheet Name: Gavin Delphia**

Assessments

Criteria Self Instructor Points

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| JUnit tests for each shape’s area method | | **5** |  | **6** |
| JUnit tests for each shape’s contains method | | **5** |  | **6** |
| Found and fixed errors (sorry, but we can’t tell you how many errors there are ☹) | | **8** |  | **8** |
| **Subtotal:** | | **18** |  | **20** |
| **Adjustments** | **Vague/Missing Documentation:** | **−0** | **−** | **− 2** |
| **Submission Requirements Not Followed:** | **−0** | **−** | **− 2** |
| **Late Penalties:** | **−0** | **−** | **25/50/75%** |
| **Total w/adjustments:** | **18** |  |  |

Documentation Statement:

I did not receive help on this assignment.

Comments from Instructor: