CSE 213 – Homework 3 Rubric

Student Name: Marisa Loraas

Grader Name: Rachel Powers

Problem	Score	Total
Style Guidelines (5 points)		**5**
Submission is named `cse213_<firstname>_<lastname>_hw3.tar.gz`	1	1
Packages are named: `oop.<firstinitial><lastname>.hw3.<number>` where <number> is the problem number	1	1
Code follows Google's style guide reasonably well, and uses four-space indentation	3	3
Temperature.java		**3**
Constructor sets the protected attribute, degrees	1	1
getKelvin(), getCelsius(), and **getFahrenheit()** return degrees	1	1
toString() returns a string with the degrees to two decimal places: "13.37 Degrees"	1	1
Kelvin.java		**2**
getCelsius() converts degrees from Kelvin to Celsius	1	1
getFahrenheit() converts degrees from Kelvin to Fahrenheit	1	1
Celsius.java		**2**
getKelvin() converts degrees from Celsius to Kelvin	1	1
getFahrenheit() converts degrees from Celsius to Fahrenheit	1	1
Fahrenheit.java		**2**
getKelvin() converts degrees from Fahrenheit to Kelvin	1	1
getCelsius() converts degrees from Fahrenheit to Celsius	1	1
WindChill.java		**4**
windChill() takes any Temperature object	1	1
windChill() returns a Fahrenheit object with the value: 35.74 + 0.6215 * F - 35.3225 * v ^ 0.16	1	1
main() prompts for a temperature, unit, and wind speed; and calls a different constructor (Kelvin, Celsius, or Fahrenheit) depending on the user input	2	2
Author.java		**13**
One constructor directly sets lastName, firstName, and middleInitial	1	1
The second constructor splits the input into two or three sub strings:	5	5

1. If there are only 2 names, the middleInitial is set to "" 2. If there 3, the middleInitial is set to the first letter of the middle name, capitalized and with a period: "D." In either case, firstName is set to the first string, and lastName is set to the last string		
getCitation() returns a string with the format "<lastname>, <firstInitial>. <middleInitial>."	1	2
splitAuthors() handles three cases: 1. Input string is comma separated: split by "," and strip "and" from the start of the last string 2. Input is separated by one "and": split by the "and" 3. Otherwise, create a singleton array Each author's name is passed to the Author constructor, and the method returns an array containing each Author object	5	5
Citation.java		**5**
Constructor sets three protected attributes: author, title, and year	1	1
Constructor sets all attributes; use splitAuthors() to set an array of authors from an input string	2	2
toString() returns a string with the format: "<author 1>,...<author N>, <title>,.<year>"	2	2
Book.java		**3**
Class extends Citation	1	1
Constructor sets author, title, year, and publisher	1	1
toString() prints a citation with every attribute in the class	1	1
Chapter.java		**3**
Class extends Book	1	1
Constructor sets author, title, year, publisher, chapterName, chapterNumber, startPage and endPage	1	1
toString() prints a citation with every attribute in the class	1	1
Journal.java		**3**
Class extends Citation	1	1
Constructor sets author, title, year, journalName, volume, startPage, and endPage	1	1
toString() prints a ciation with every attribute in the class	1	1
Newspaper.java		**3**
Class extends Citation	1	1
Constructor sets author, title, year, newspaper, month, and day	1	1
toString() prints a citation with every attribute in the class	1	1
Website.java		**3**

Class extends Citation	1	1
Constructor sets author, title, year, url, month, and day	1	1
toString() prints a citation with every attribute in the class	1	1
Bibliography.java		**2**
main() creates an array of Citations, containing at least one Book, Chapter, Journal, Newspaper, and Website	1	1
main() uses a loop to print each citation in the array	1	1
Shape.java		**2**
Class is abstract, with abstract methods area(), perimeter(), and inBounds()	1	1
Rectangle and Circle both extend Shape	1	1
Parallelogram.java		**18**
The default constructor calls super() and sets skewAngle to pi/2	1	1
The second constructor calls super(pt1, pt2) and sets skewAngle to the given value; if the given angle is greater than pi, sets the skewAngle to pi/2	2	2
getLowerRight() returns the Point: 1. super.getLowerRight() if skewAngle = pi/2 2. (upperRight.x - height / tan Θ_{skew}, y) otherwise	1	3
getUpperLeft() returns the Point: 3. super.getUpperLeft() if skewAngle = pi/2 4. (lowerLeft.x - height / tan Θ_{skew}, y) otherwise	1	3
area() returns *width* * *height* where *width* is the distance between the lower left and lower right points (or the upper left and upper right points)	3	3
perimeter() returns the sum of the distances between adjacent points	3	3
inBounds(pt) checks that the given point is in bounds for the corresponding rectangle (super.inBounds()), and somehow checks that pt is between the lines of the left and right edges.	3	3
Ellipse.java		**17**
Default constructor calls super() and sets minorRadius equal to radius	1	1
Second constructor sets the center point, radius, and minorRadius	1	1
focalDistance() returns $\sqrt{\lvert r_1^2 - r_2^2 \rvert}$	3	3
eccentricity() returns $f/\max(r_1, r_2)$	3	3
area() returns pi * r_1 * r_2	3	3
perimeter() returns pi * $\sqrt{2}$ * $\sqrt{(r_1^2+r_2^2)}$	3	3
inBounds(pt) returns true if and only if: $(pt.x - center.x)^2/r_1^2 + (pt.y - center.y)^2/r_2^2 \leq 1$	3	3
JUnit Testing (10 points)		**10**
A test class that has 9 different tests about Parallelogram.java (.5 pts each)	4.5	4.5

A test class that has 11 different tests about Ellipse.java (.5 pts each)	5.5	5.5
Total Score	95	**100**
Comments:		