

COS125 - Precept 3 (Conditionals)

1 Truth Tables

Write the truth tables for the following boolean expressions.

a	b	a && b
false	false	
false	true	
true	false	
true	true	

a	b	(!a) b
false	false	
false	true	
true	false	
true	true	

a	b	c	a && (b c)
false	false	false	
false	false	true	
false	true	false	
false	true	true	
true	false	false	
true	false	true	
true	true	false	
true	true	true	

2 Quadrants of the Cartesian plane

Please write a program `Quadrants.java` that takes two `double` command-line arguments, x and y , corresponding to the coordinates of a point in the Cartesian plane, and prints the quadrant that contains the point (x, y) .

We consider the x and y axes to not belong to any quadrant. Thus, if (x, y) lies in one of the axes, your program should identify this and print accordingly; and likewise if (x, y) is the origin $(0, 0)$. See Figure 1 for a picture of the quadrants.

The following table shows examples of possible inputs to `Quadrants.java` alongside the expected output:

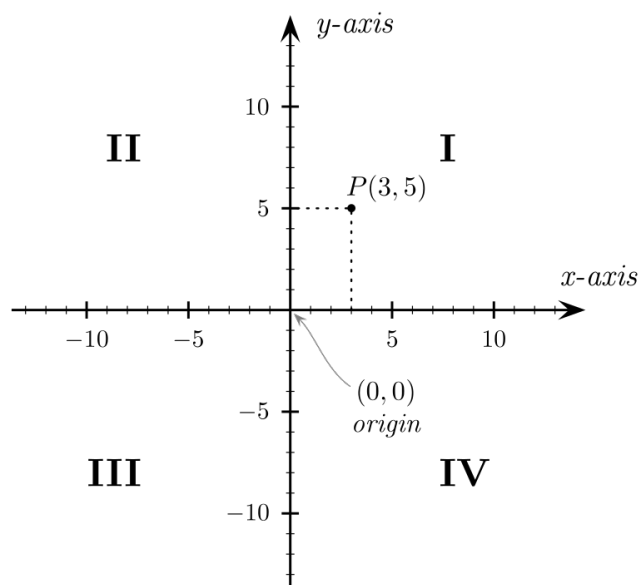


Figure 1: Quadrants of the Cartesian plane.

x	y	Output
0.0	0.0	Origin
1.0	0.0	x-axis
0.0	12.5	y-axis
12.5	12.5	Quadrant I
42.0	-12.5	Quadrant IV

3 Wordle Feedback

In the game of [Wordle](#), you try to guess the 5-letter word of the day. When your attempt is wrong, the game tells you which letters you guessed correctly.

We will create a program `WordleFeedback.java` that, in a similar scenario, gives a more limited form of feedback. The word of the day should be stored in the variable `WORD_OF_THE_DAY`, and in this example will have the value "SHARE".

Our program will receive one command-line argument corresponding to the user's current attempt. It should then print the letters that were guessed correctly in the correct position, and print a dash otherwise. For example, if the user guessed `SNAIL`, our program should print `S-A--`; and if the user guessed `PHASE`, our program should print `-HA-E`.

You may want to use `s.charAt(index)`, a Java method that returns the character at position `index` in the string `s`.

4 Magic 8 ball

A magic 8 ball toy is one that looks like a billiard eight-ball, but when you shake it, it shows various answers to yes and no questions. For instance, if you ask it “Will I conquer my programming demons?” it might respond with “Without a doubt.” Assume that our magic 8 ball uses a 6-sided die inside that holds all the answers. Write a program called `Magic8Ball.java` that when executed, displays a random answer from the metaphoric 6-sided die. You can choose the answers your 8 ball can display. (Hint: you may want to use `Math.random()`)

Now, create a second version of this program called `Magic8BallInteractive.java`. This program should take in a question from a user as a single command line argument. The question should be formed as a yes or no question. For instance, “Will I” and “Should I” questions lead to yes or no answers, but “Who”, “What”, and “Where” do not. If your program determines that the question is not in an appropriate format, print “Question is not in a valid format.” Otherwise, display a random answer from your magic 8 ball.

Do you prefer one version of the program to another? Which one and why?

5 Rock, Paper, Scissors

Write a program that simulates a Rock, Paper, Scissors game, where a user is one player and the computer is the second player. The program should take in one of “rock”, “paper”, or “scissors” as the user choice in the game. It should then simulate the computer choosing one of the three options by using the `Math.random()` method. After the computer makes its choice, the winner is determined by comparing the user and computer decisions:

- Rock beats scissors
- Paper beats rock
- Scissors beats paper

Your program should print either “Computer wins!”, “You win!”, or “Tie!” based on these rules.

Credits

media	source	license
cartesian plane	Wikipedia	CC BY-SA 3.0