

Assignment week 3

Code located at: <https://github.com/shahhaard47/JFK-JavaAssignment>

USING METHODS and CONDITIONALS

Initial Setup

1. copy the following code into a file called `SwagBot.java`
2. fill in the blanks to make the code run
 - edit implement code where the comments instruct you to do so
 - change to java that should go there

FILE: `SwagBot.java`

```
class SwagBot {

    // this method should return which direction robot is currently heading
    public static <returnType> calculateRobotDirection(double leftSpeed, double
rightSpeed) {
        String direction = "";
        // check if robot is turning right
        if (leftSpeed > rightSpeed) {
            direction = "right";
        }
        else if(/* code */) {
            dierction = "left";
        }
        else if(/* code */) {
            direction = "fwd";
        }
        else {
            direction = "bkwd";
        }

        return direction; // this return whatever `direction` is back to whoever
(whomever?) called it
    }

    // should print summar of data passed in as parameter
    /* example output:
        Robot name: SwagBot, Drive type: tank, current driving direction: fwd
        Details: leftspeed = 0.5, rightspeed = 0.5
    */
    // Notice 'void' in function header, this means function does not return
anything
    public static void printDashboard(double name,
                                     double leftSpeed,
                                     double rightSpeed,
                                     String dir,
                                     String driveType) {

        // put code here
        // first line

        // second line
    }
}
```

```

    }

    // write a function named "betterDashboardPrint" that does not require "String
    dir" parameter
    // hint: inside your new betterDashboardPrint you can call the
    calculateRobotDirection to get the direction

    public static void main(String[] args) {
        final String robotName = "SwagBot";
        double leftSpeed = 0.0; // should be between [-1.0, 1.0]
        double rightSpeed = 0.0;
        String direction = null; // 'left', 'right', 'fwd', 'back'
        String driveType = "tank"; // either 'tank' or 'arcade'

        // call printDashboard to print current state of robot
        direction = calculateRobotDirection(leftSpeed, rightSpeed);
        printDashboard(robotName, leftSpeed, rightSpeed, direction, driveType);

        // make sure robot is stopped
        leftSpeed = 0; rightSpeed = 0;
        direction = calculateRobotDirection(leftSpeed, rightSpeed);

        // go forward with full speed
        leftSpeed = 1.0;
        rightSpeed = 1.0;
        direction = calculateRobotDirection(leftspeed, rightSpeed);

        // write code to go fwd with half speed
        leftSpeed = //code
        rightSpeed = //code
        direction = // call calculateRobotDirection

        // write code to go bkwd with full speed -1, -1

        // write code to turn right

        // write code to turn left
    }
}

```