Name: _____ **Basics Gradesheet** MCS-178

_____ **20 Checkerboard**

_____ [10] generates boards correctly for all positive integer n (not just when n equals 4 or 5)

_____ [5] checks for the 1 command line argument condition and gives error message if condition not met

_____ [5] uses only one nested loop and uses only one if statement within the nested loop

_____ **20 Bear**

_____ [10] bear face is well-proportioned and looks nice

_____ [5] uses `ellipse`, `line`, and `arc` appropriately

_____ [5] calls shape-drawing functions in the correct order; no unnecessary calls

_____ **20 Bears**

_____ [5] `bear1` works according to specifications

_____ [5] `bear1` code modifies every shape-drawing calls of `Bear.kt`

_____ [5] calling `bear2` gives exactly the same output as `bear1`

_____ [5] `bear2` uses `tranlate` and `scale` but retains all shape-drawing calls of `Bear.kt`

_____ **20 Random Walk Simulation (a)**

_____ [10] simulation shows graphically and stops after exactly `n` steps

_____ [5] checks for the 1 command line argument condition and gives error message if condition not met

_____ [5] computes the squared distance correctly and outputs it at the end of simulation

_____ **20 Random Walk Simulation (b)**

_____ [10] performs all the simulation experiments correctly and calculates the mean squared distance correctly

_____ [5] checks for the 2 command line arguments condition and gives error message if condition is not met

_____ [5] correct hypothesis on how the mean squared distance grows as a function of `n`

Note. Don't forget to fill out and submit the `readme.txt` file together with the project's source files.

_____ **TOTAL**