

## PRELIMINARY PROGRAMMING STYLE CHECKLIST

Before your program can be considered complete, you must ensure that the following requirements are met. Please ask the instructor if there is anything you are unsure about.

1. Code is properly commented – Meaningful comments only. Do not state the obvious. Examples:
  - i. meaningful : `//follow the line until you see silver.`
  - ii. Useless: `//start a program, or // sensor value higher 50, or //stop motors.`
- 2) No “X’s” remaining in the compiler errors window – pay attention to proper syntax, and fix all warnings. Even if the warnings are for unreferenced functions or variables, try your best to minimize them.
- 3) Code is neat and properly formatted, using appropriate curly braces and indentations – Each curly brace should have its own line. Take advantage of the “fix formatting” button on the toolbar!
- 4) Header files should only contain variable and sensor initializations. It is a good idea to create a header file if you do not already have one, especially if you are working as a team.
- 5) All variables should either be initialized in a header file or at the beginning of each function (Note: even `task main()`). Do not initialize variables randomly throughout your code.
- 6) Variables should have meaningful names that reflect their functionalities. Try to limit a single letter only to represent something like a local loop count within “a small scope”. For large scope, even the loop count variable should be more intuitive.
- 7) Always create a variable or macro for a repetitiously used constant.
- 8) Minimize global variables.
- 9) Prefix the global with 1st letter to be in upper case. If name is long, use camel style, e.g. `EncoderPerDegee`.
- 10) Must use a proper variable name for your sensors and motor ports. Do not directly reference it as `S1, motorA, etc.`, inside functions.

Exercise:

Have students to make 2 examples for each point. One bad and one good example.