18-642 Recitation #2

Sept. 7, 2019
Updates

• Homework:
  – Look for homework grades on Canvas
    • Feedback will be in the comments of your submission
    • Goal: assignments graded within 1 week of submission

• Fill out weekly survey every week!
  – Starting with week 3, access to next Canvas week is blocked until you fill out the survey.
    • If you’re trying to work ahead let us know for a workaround.
    • This is intended to avoid problems in past semesters with students forgetting to fill out the surveys in a timely manner
Updates

- Projects:
  - Project 2 due tonight
  - Project 3:
    - Released; due in one week
  - Expect projects hand-in every week, due Friday night
    - Expectation: hand-in by 5 AM Saturday morning
    - Canvas deadline will not be extended due to last minute connectivity issues, etc. Don’t wait until the last minute!
Updates

• Reminder to use staff email and office hours for questions
  – Don’t use “canvas messages”
  – Emails ece642-staff@ instead of individual instructors/TAs
  – Office hours are posted as a canvas announcement
    • Watch for location changes
Today

- Project #2 review
- Project #3
- Homework presentations
Project #2 Questions?

• You’re graded on process and effort
• Successful implementation **not required**
• Please follow the time limits
  – 1hr for first implementation attempt
  – 2-6 hours for clean up
  – Up to 1hr for second implementation attempt
Project 3

- Change code style according to a checklist
- Asks you to use a physical copy of the checklist
- Reflect on the checklist vs changes you made in Project 2
- Spend some time making it nice
  - You will have to live with this code later!
  - Don’t slack off!
1) Variables and procedures have minimum scope
   It is fine to define all variables at the front of the procedure
   Defining at smaller scope (such as within a “{ }” block) is at your discretion

2) All variables are automatic (no globals; minimal use of “static” keyword)
   Allocated on the stack unless required to be permanently allocated

3) All variables use strongest and simplest type possible, with no floats
   Add extra typedefs at your discretion

4) All base types are from types.h
   Variable size is at your discretion within reason

5) Geometric pairs (e.g. (X,Y)) are coupled using typedefs

6) #define is not used

7) “Magic numbers” are not used
   Use const for single values and enum for sets of related values

8) Switch statements are used to decide among enum values rather than if/else if

9) Every switch statement has a default error handling clause
   Default activates ROS_ERROR

10) Every variable has a meaningful name
    Does not require explanation to someone else

11) Code is commented
    At a minimum, comment each function: purpose, inputs, outputs, saved internal state

12) Code conforms to Spaghetti Factor guidelines on a per-procedure basis
    Number of globals: ___ SCC*: ___ SLOC/20*: ___ ⇒ SF*: ___
    *in most complicated module/procedure that has the highest SF
    Be ready to defend complexity above SF=10 as being absolutely necessary

13) All the code is in a single .cpp file
    (This changes later; for now this is to ease review & grading)

14) All conditionally executed statements are enclosed by “{ }”

15) All indentation and similar style is consistent

16) No copy-pasted code is present
    Blocks of code that have the same functionality are factored out into modules
Project #3 Questions?

• Spaghetti Factor is in the Global Variables on-line lecture
• Note: you get three free “misses” on checklist this week
  – But you’ll need to fix them next week!
• Questions about what a checklist item means?
Other Questions?
Student Presentations

• HW #3: Failure Stories