

# ICS 211 Assignment 13

## Pedagogical Code Review (PCR)

**Assignment 13 is an in-class assignment to perform Pedagogical Code Reviews (PCR) and Black Box Testing (BBT) for 2 assignments of your choice (from 1-4 or 6-11).** For this assignment, you need to do the following:

1. Choose 2 assignments, 1 for the Pedagogical Code Review and 1 for the Black Box Testing
2. Put assignment #'s and names of your code choices in Anthony's Github site or send email to your TA **by 11:59, Saturday December 7th**
3. For the PCR, **bring a printout of your chosen code** on the designated code review dates which are:
  - Section 1 **Tuesday, 12/10 Lecture 10:30am**
  - Section 2 **Tuesday, 12/10 Lecture 10:30am**
  - Section 3 **Wednesday, 12/11 Lecture 3:00pm**
4. Exchange email addresses with fellow group members (same group members for PCR and the BBT)
5. Perform face to face Pedagogical Code Reviews in the Lecture (BBT will be done in the Lab)
6. Email the inspection logs generated from the Pedagogical Code Review to the team members and TAs by 11:59 on the evening of the reviews.
7. The grading of the PCR portion is based on the artifacts and documentation created during the reviews.
7. You may resubmit reviewed assignment solutions for a new grade by **11:59, Friday December 13th**
8. Your final assignment grade will be an average of your original and new grade

### What is an Industry Code Review?

Industry Version Code Reviews involve the following five general steps:

1. Plan the inspection of a specific piece of code.
2. Meet with an inspection team to distribute the code to be inspected and set inspection goals.
3. Have members of the inspection team inspect the code for defects on their own time.
4. Hold an inspection meeting to log issues found inspections and to discuss additional issues.
5. Edit the code to address the issues uncovered in the inspection, and check that the issues have been resolved.

### What is a Pedagogical Code Review?

Pedagogical Code Reviews differ from formal industry code reviews because they need to be implemented within lecture and laboratory sessions. Thus, there is not time for an initial group meeting (step 2 above), nor can students be expected to have reviewed each other's code prior to the face-to-face inspection sessions (step 3 above) instead students should come to the face-to-face sessions **with printouts of their assignment solutions**, so that the assigned teams can perform their inspection work collaboratively during the face-to-face inspection sessions. Students are then able (a) to edit their assignment solutions in response to the feedback they receive in the code review process, and subsequently (b) to re-submit their assignment solutions for a new grade (step 5 above). For the purpose of Assignment 13, students in each section are divided into assigned teams of 3-4 members.

### Pedagogical Code Review Procedure

The course assignments that are the focus of the reviews vary in complexity and length but you may pick 2 assignments of your choice, from 1-4 or 6-11 to be reviewed (just email your choices to your TA by 11:59, Saturday December 7th). Teams review their members code against coding best practices augmented with a list of requirements for the specific programming solution being reviewed. The best practices can include questions in the seven general categories shown as an example in Table 1.

In the PCR assignment, teams review the code of all of their members. Individual team members rotate through the following well defined roles for each individual review: If a team has only 3 members, the *Author* also assumes the role of the *Reader*.

- *Author*—the person whose code solution is being inspected; may be called upon to clarify his or her code as needed.
- *Reader*—reads the Author’s code solution aloud.
- *Inspector*—takes a leading role in inspecting each line of code against the provided code review checklist.
- *Recorder*—documents issues that arise in the inspection in a provided inspection issue log.

Each individual code review follows a well-defined procedure that the Moderators (TAs,LAs and Instructors) check. The Moderators assign teams and their rotating roles to team members. To set the stage for the review, the following scenario is assumed:

The author has created this solution and asked us to help make it better. Please focus your comments on improving the solution. Look beneath the superficial minor defects or style issues you see, to hunt out **significant defects**. If you aren’t sure if something is a defect, point it out and we’ll decide as a team. Our goal is to identify defects. In general, after about one minute of discussion on an issue, it should be resolved. If not, record it and move on.

At this point, the Reader begins reading the Authors’ code. Code is most often read line-by-line, but the Reader is free to summarize sections of code if line-by-line inspection isn’t necessary. As lines of code are read, anyone on the team can raise issues for discussion. The team is required to classify issues both in terms of severity (minor, moderate, or major), and in terms of their checklist categories (such as one of the seven general categories presented in Table 1). The Recorder carefully documents each issue that is raised and discussed by the team.

At the end of a given review, the Moderator asks the Recorder to read aloud all issues, in order to make sure that they are logged properly and that the team is in agreement. The entire inspection process is repeated until the code of all team members is inspected. Finally, copies of the inspection logs are emailed to the Author, the other team members and the TAs by 11:59 on the evening of the PCR.

**Table 1. Code inspection issue categories**

#	Issue Category	Sample Question
1	Code meets assignment requirements	Is code implemented correctly according to specifications?
2	Structure and design	Does the code adhere to proper object oriented design?
3	Documentation, standards, and formatting	Are all comments consistent with the code?
4	Variables and constants	Are all variables properly defined in the smallest possible scope?
5	Arithmetic operations	Are divisors tested for zero?
6	Loops and branches	Are loop termination conditions obvious and invariably achievable?
7	Defensive programming	Is the code using uninitialized array locations?

# Black Box Testing (BBT)

**Email the code you chose for Black Box Testing (BBT) to your fellow group members (the same group members you worked with during the PCR test) so everyone in your group has access to the working compiled code.**

**Bring to your lab 3 additional tests (that were not included in the JUNIT testing) for each program chosen by the other members of your group**

The designated code review dates are:

Section 1 **Tuesday 12/10 Lab 12:00pm**

Section 2 **Wednesday 12/11 Lab 12:00pm**

Section 3 **Wednesday 12/11 Lab 4:30pm**

The BBT is performed on the second program that each individual team member chose. These programs will be listed in Anthony's Github area or emailed by your TA. In BBT reviews, team members do not have access to the code so there are no reader and inspector roles. Each team member tests the other team member's code with the JUNIT tests plus the 3 additional tests. The BBT log is a description of any tests that are performed. Any tests used from JUNIT should also be described. Copies of the BBT logs should be emailed to the Author, the other team members and the TAs by 11:59 on the evening of the BBT. You may re-submit the BBT reviewed assignment accompanied by a description of any changes made for a new grade by **11:59, Friday December 13th**. Your final assignment grade will be an average of your original and new grade.