


[Dashboard](#) > [Training](#) > ... > [Course Catalog](#) > [Object Boot Camp](#) 

# ThoughtWorks®

## Object Boot Camp

Training

Welcome [Luca Minudel](#) | [History](#) | [Preferences](#) | [Log Out](#) [View](#)[Attachments \(10\)](#)[Info](#)[Review](#) [Browse Space](#)Added by [Paul Gross](#), last edited by [Amiruddin Nagri](#) on Aug 09, 2012 ([view change](#))Labels: [session-plan](#), [object-oriented](#), [programming](#), [design](#), [test-driven](#), [development](#), [thoughtworks-university](#), [pair-programming](#) [EDIT](#)

### Instructor's Note

Object Boot Camp is currently offered in two variants:

1. a three week full time course delivered at client sites as part of Organizational Transformation (OT) initiatives
2. a shortened course presented during several shorter session during ThoughtWorks University.

These pages are an attempt to collect the course material in one place, while highlighting the differences in delivery between the two variants. Where possible, the three week course has been used as the baseline (Standalone Syllabus), with a separate tab available to point out these differences.

[Standalone Syllabus](#)[University Syllabus](#)[QA Bootcamp Syllabus](#)

The following material is structured by learning activities, [i.e] exercises in which the participants demonstrate their understanding of the principles that are called out during the course. As presented, this will be most familiar to those that have participated in and/or delivered the course in the past. If you're interested in seeing how the activities are arranged by key OO principle, see the [University Syllabus](#) tab.

## Sessions

### Part I - Object Principles and Agile Practices

1. [Introduction](#)
2. [Chance](#)
3. [Quantity](#)
4. [Bester](#)
5. [Recursion](#)
6. [Collections](#)
7. [Graph - Can Reach](#)
8. [Graph - Hop Count](#)
9. [Graph - Cost](#)
10. [Graph - Path](#)

### Part II - Design Patterns

1. [Introduction to Design Patterns](#)
2. [Parking Lot - Observer](#)
3. [Parking Lot - Composite](#)
4. [Parking Lot - Template Method](#)

5. [Parking Lot - Visitor](#)
6. [Parking Lot - Adapter](#)
7. [Refactoring to Patterns](#)
8. [Design Pattern Conclusion](#)
9. [Course Wrap-up](#)

## Daily Activities

---

### Nightly Preparation

The most important preparation that facilitators do once OBC is underway is to write the next day's code the night before. This ensures that the problems and discussions for the next day are fresh in their minds.

### Daily Morning Safety Check

- Hand out post its and students write 1-5 (say anything, say almost anything, middle of the road, not say much, say nothing).
- Gather the postits and use the information to tune your teaching. For example if people are feeling unsafe ask for volunteers to show code as opposed to choosing 'volunteers'
- Do this until the class feels safe (but not the first day). Also do it before each end of week retrospective.

### Daily Morning Review

- Purpose
  - Facilitator can assess students progress
  - Students can realise how much they have learned - this is important given their daily struggle
- Activities
  - Review vocab by asking one student to pick a word and another student to define it. A variation on this is to have students who have not yet defined a word to stand, this makes it easy to see who has yet to define something.
  - Ask students what the weirdest thing they heard the day before was.
  - Ask students what they learned the day before. Go into details of the coding that they did. For example *How did the Quantity solution develop yesterday?*

### End of Day / Week Retrospective

- Safety check
- Retrospective Options [Retrospective Summary Template](#) 📄
  - What went well / What to do better
  - Good / Bad & Ugly / Learned / Wanted to Learn (But Didn't) / Pre-requisites
  - Good / Bad \* Ugly / Learned / Wanted to Learn (But Didn't) / Describe the Class to Others
- Emotional seismograph - check how the students are feeling. The class is tough on students. Often the class will start with an immediate dip for several days followed by a recovery. It is important to monitor this and make sure the students do recover and feel knowledgeable and confident. Otherwise the learning can be lost with unhappy students discarding it as a waste of time.

## Things to Look For throughout OBC

---

- Students running only recent tests instead of all tests.
- Students not clearly distinguishing refactoring, writing tests and writing code. They may refactor on a red bar, or write code on a green bar.

## Useful Exercises

---


### The 3-sided hat:

- When: This is useful when students are not clearly distinguishing between refactoring, writing tests & writing code
- What: A 3-sided hat with the words refactoring, writing test & writing code written on the sides in the appropriate color: refactoring = green bar, writing tests = green bar & writing code = red bar. The students wear the hat and turn it to the side represent what they are doing at the moment.

## Client Overview of OBC

[OBC Overview](#) - This can be used if clients ask for an overview of Object Bootcamp. Read it over to make sure you are actually covering everything mentioned before sending it out.

## Equipment Required

- room
- projector
  - SVGA extension cord (optional - to make it easier for students to plug into the projector)
- whiteboard
- 2 flipcharts
- markers
- timer
- notecards / postits (for retrospectives)
- flash drives (or other method to copy code for showing / rebaselining)
- pairing stations
  - power for machines (powerstrips, etc)
  - same IDE installed
  - same language installed
  - separate keyboards & mice if laptops (optional - to make pairing easier)
  - separate monitors if laptops (optional - to make pairing easier)
- [DEMO:handouts](#)
  - synopses of sessions
  - laminated hotkeys cheatsheet
  - talk to Patti Hall about printing & preparation

## » Access Groups

The following people have edit access to this set of pages. (Please, submit a request to 8600 to have yourself added to this group.)

 [36 children](#) | [View in hierarchy](#)

 [2 comments](#) |  [Add Comment](#)