# HtDF Design Quiz

NOTE: to get the most out of this design quiz you should do it alone. Please don't discuss it on the forums, or answer other people's questions about it on the forums until after the submission deadline passes and the self-assessment period starts.

This design quiz will have 3 phases:

- In step one you will design a solution to a given problem, and turn that solution in using the edX Peer Assessment tool.
- In step two the instructor will release a video showing the assessment of several randomly selected submissions.
- In step three you will do a self-assessment of your solution.


PROBLEM:

Design a function that consumes two images and produces true if the first is larger than the second.

Note that limitations of the edX assessment tool mean that you cannot use images directly in your .rkt file. You can of course use image primitives like rectangle. But you cannot paste actual images into your .rkt file for this quiz, even though you normally can do that in BSL. Furthermore, unfortunately the assessment tool does not preserve the indentation of your code. When it comes time for the self-assessment, you can assess your original .rkt file, or copy your submission back into DrRacket and press cmd/ctr-I.

Your design will be assessed using the following rubric:

1. Is the program "commit ready"?

The file should be neat and tidy, no tests or code should be commented out other than stubs and templates and all scratch work should be removed. The identation should conform to course conventions and typing CMD-I (CTL-I on Windows) should not move anything. (But note that limitations of the peer tool mean that indentation can only be self-assessed.)

2. Is the design complete?
   All HtDF design elements should be present, and each element should be well-formed.
3. Does the design have high internal quality?
   The signature should be correct, the purpose should be clear and succinct, the examples should be sufficient to test and explain the function. The function name should be well chosen and should describe what the function does, not how it does it. The stub should match the signature. The template should be correct. The function body should be clear. When the program is run all the tests should pass, and those tests should cover the entire program.
4. Does the design satisfy the problem requirements?
   The function design should satisfy the problem statement. If there is any ambiguity in the problem statement the function design should identify and resolve that ambiguity.

---

Please watch the following video before completing your self-evaluation. We go through several different submissions and walk through how we would apply the rubric. Keep in mind that these are only a few example solutions. If you resolved the ambiguity in the problem statement differently, your solution could still be correct. The video is not meant to provide you with the answer, but rather to show you how to evaluate your solution systematically using the rubric.

---

## Evaluation Tutorial Video

(Caption will be displayed when you start playing the video.)

**Video**
Download video file

DESIGN QUIZ 1

**Status**

This assignment has closed. One or more deadlines for this assignment have passed. You will receive an incomplete grade for this assignment.

▸ **Your Response** due Jun 18, 2015 06:00 WIB (in 0 minutes) ⚠
INCOMPLETE

▾ **Assess Your Response** due Jun 27, 2015 06:00 WIB (in 0 minutes)
⚠ INCOMPLETE

**Status**

The due date for this step has passed. This step is now closed. You can no longer complete a self assessment or continue with this assignment, and you will receive a grade of Incomplete.

▸ **Your Grade: Not Started**