

# Exam Preparation

JavaScript OOP – July 2014

---

**Telerik Software Academy**

Learning & development

<http://academy.telerik.com>

- ◆ The exam will take exactly 3 hours in two groups:
  - ◆ At 10:00 on Mon, 7-Jul-2014
  - ◆ At 19:00 on Mon, 7-Jul-2014
- ◆ The exam will consist of a single task testing your expertise with:
  - ◆ Understanding and creating function constructors
  - ◆ Using modules and Require.js

# Sample Task #1

- ◆ You must complete the functionality for a TODO list
  - ◆ A TODO list has a single container
    - ◆ A container can contain only sections
      - ◆ Sections can be zero or more
  - ◆ A section have a title (string) and only contain items
    - ◆ Items can be zero or more
  - ◆ Items have content (string)

# Sample Task #1 (cont)

- ♦ Implement the methods for the types Item, Section and Container
  - ♦ You are allowed only to extend the scripts inside the files 'container.js', 'item.js' and 'section.js'
- ♦ Methods descriptions:
  - ♦ Container.add(section) adds a section to the container
  - ♦ Container.getData() returns a JSON array with sections
  - ♦ Section.add(item) adds an item to the given section
  - ♦ Section.getData() returns its title and an array with items
  - ♦ Item.getData() returns the content of the item
- ♦ You are given a few files, that must not be changed

# Sample Task #1

## Live Solution

# Sample Task #2

- ◆ You are given a courses database
  - ◆ Courses have title and students that participate in them
    - ◆ Courses must be able to return the best students either by their exam score or by their total score
    - ◆ The score of a student for a course is calculated by a given formulate, that is specific for each course
  - ◆ Students have exam score, homework score, attendance score, teamwork score and bonus score
    - ◆ Each of these components are part of the total score of the student for a course
  - ◆ Each student can be part of zero or one course



# Sample Task #2 (cont.)

- ◆ You are given a courses database
  - ◆ By the given files, implement the functionality of the student.js and course.js
    - ◆ You are only allowed to change the contents of these two files
    - ◆ Create them using RequireJS

# Sample Task #2

## Live Solution

# Questions?

