University of Glasgow | School of Computing Science

# Assessed Coursework

| | | | | |
|---|---|---|---|---|
| Course Name | CBSE exercise 3 | | | |
| Coursework Number | | | | |
| Deadline | Time: | 4.30pm | Date: | 13th March 2015. |
| % Contribution to final course mark | 20% | | This should take this many hours: | 4hrs/week |
| Solo or Group ✓ | Solo | | Group | X |
| Submission Instructions | Submission is via Moodle | | | |
| Marking Criteria | Task based | | | |
| Please Note: This Coursework cannot be Re-Done | | | | |

## Code of Assessment Rules for Coursework Submission

Deadlines for the submission of coursework which is to be formally assessed will be published in course documentation, and work which is submitted later than the deadline will be subject to penalty as set out below. The primary grade and secondary band awarded for coursework which is submitted after the published deadline will be calculated as follows:

(i) in respect of work submitted not more than five working days after the deadline
  a. the work will be assessed in the usual way;
  b. the primary grade and secondary band so determined will then be reduced by two secondary bands for each working day (or part of a working day) the work was submitted late.

(ii) work submitted more than five working days after the deadline will be awarded Grade H.

Penalties for late submission of coursework will not be imposed if good cause is established for the late submission. You should submit documents supporting good cause via MyCampus.

**Penalty for non-adherence to Submission Instructions is 2 bands**

**You must complete an "Own Work" form via https://webapps.dcs.gla.ac.uk/ETHICS for all coursework UNLESS submitted via Moodle**

# CBSE 2014/2015

## Exercise 3

During the next 8 weeks, you will work in groups to implement a platform for running component-based software. First you will propose a lightweight (network topology agnostic) component model, and then implement a framework based on the model. The behavior and structure of the model shall be specified using Role Activity Diagrams (RAD) and/or UML component diagram. Your product shall be assessed based on achieving the following tasks:

### Task 1:

The model shall enable component interaction based on the following requirements:

1. By using a proxy, a component shall be able to make a remote method invocation on another component. Communication between components shall be enabled using UNIX sockets IPC. You should not use Java RMI for this exercise. (3 marks)
2. Message passing during a remote method invocation shall be encoded as binary or text messages. (3 marks)
3. The model shall provide a mechanism for associating metadata to encoded messages. (3 marks)

### Task 2:

The model shall provide the following compendium services using program reflexive capabilities:

1. Component service advertisement and discovery. (3 marks)
2. Component lifecycle management. (3 marks)
3. Stateful, stateless and persistent session management. (3marks)
4. **OPTIONAL:** Security management via access control (verifying access rights) and authorization (granting access rights). (2 marks)

### Task 3:

1. Demonstrate interaction between components based on implemented framework. (2 marks)

### Deliverables:

At the end of the exercise you will submit the following on Moodle:

1. A document describing your model and framework (developer/user manual). The document should be short (10 pages to 20 pages (exceptionally ~30 pages), and should include:
    a. A very short introduction outlining the content of your report.
    b. A description of the general architecture (behavior and structure), with an illustration of how components interact with each other on the platform to achieve a common objective.
    c. Key modelling decisions, including the fundamental data structures for achieving each task items highlighted above.
    d. Basic classes and their non-private methods, together with brief descriptions of them (important ones would be given long descriptions; while trivial ones very short descriptions).
    e. Conclude with observations on your model and framework, which may include a comparison with other existing designs such as OSGi, EJB, Java RMI, FIPA, Corba, OpenCOM, Fractal etc.

    The document as a whole should make clear how an invocation of the form:

    *a = r.method(x,y,...);*

    from a component A, can be compiled and act as a remote method invocation on a component B. and how the different compendium services in Task 2 can be used to regulate the invocation. Here, *r* is a your remote object in component B residing in a different VM (actually, *r* is presented as a proxy to component A, but conceptually it exist remotely in B).

2. A jar file (and code) containing the implementation of the framework.
3. At least two components (jars and code) that can be used to test component interaction using the framework.

**Datelines:**

- Group presentation/demonstration **Monday 9th March 2015**.
- Submission of deliverables **Friday 13th March 2015**.

**Process:**

Each group should meet at least once a week to discuss and assign tasks, as well as evaluate progress. There will be four formal retrospectives that will take place during lab/tutorial sessions in week 20, 21, 22 and 23 respectively. The course tutor will be available during these sessions to provide some guidance where needed. At the end of each retrospective, you are expected to submit a short progress report stating the following:

1) Tasks that went well, or did not go well in the past week.
2) Mitigation actions that will be carried out to ameliorate tasks that did not go well.
3) Plans for the next week.

During weeks 18 and 19, you are still expected to meet, work on the exercise, and carry out retrospectives to evaluate your progress. During this period, you will build background knowledge, and iterate over your initial design architecture. The course tutor is also available during lab/tutorial sessions to provide feedback. Over the duration of the exercise, you are encouraged to do incremental writing of your final report as you incrementally model and implement your system (leaving your writing to the last week could be tricky!). You might also want to utilize some time in week 24 to prepare for your group presentation/demonstration that will take place in week 25.

Note: while retrospectives are not assessed, they are necessary to provide insight on contributions of individual group members as well as appropriately review your progress. It is also important as it provides the tutor some understanding on areas where your group or individual in the group may need further support.

At the end of this exercise, every individual in a group will submit an objective peer review of effort and contributions of group member to the tutor. Submission shall be made via Moodle. For the review, you will benchmark each member of the team on a Likert scale of 1-5. **Where 1 means no contribution and 5 is excellent contribution**.