GMIT
INSTITIÚID TEICNEOLAÍOCHTA NA GAILLIMHE-MAIGH EO
GALWAY-MAYO INSTITUTE OF TECHNOLOGY

**GALWAY- MAYO INSTITUTE OF TECHNOLOGY**

# SEMESTER 1 EXAMINATIONS 2016/2017

**MODULE:** COMP08011 - DISTRIBUTED SYSTEMS

**PROGRAMME(S):**
GA_KSOFG_H08 BACHELOR OF SCIENCE (HONOURS) IN SOFTWARE DEVELOPMENT

**YEAR OF STUDY:** 4

**EXAMINER(S):**

| | |
|---|---|
| Dr. John Healy | (Internal) |
| Dr. Des Chambers | (External) |
| Mr. Tom Davis | (External) |

**TIME ALLOWED:** 2 Hours

**INSTRUCTIONS:** **Answer 4 questions. All questions carry equal marks.**

---

**PLEASE DO NOT TURN OVER THIS PAGE UNTIL YOU ARE INSTRUCTED TO DO SO.**
The use of programmable or text storing calculators is expressly forbidden.
Please note that where a candidate answers more than the required number of questions, the examiner will mark all questions attempted and then select the highest scoring ones.

---

***There are no additional requirements for this paper.***

**1.** (a) Using diagrams and examples where appropriate, explain the following terms as they apply to distributed systems:

- Heterogeneity **(3 Marks)**
- Scalability **(3 Marks)**
- Transparency **(3 Marks)**

(b) *"A façade can be used to shield a client from the complexity of a distributed system and promote loose coupling between service requestors and providers."*

Discuss this statement. Your answer should address how **session** and **message façades** can be used to aggregate services and increase the scalability of a distributed system.

**(16 Marks)**

**2.** (a) *"Marshalling frameworks based on highly structured unicode formats have largely supplanted serialisation and binary data transfer formats."*

You are required to provide a critique of this statement. Your answer should compare XML, JSON and lower-level **marshalling** formats in terms of heterogeneity, extensibility and efficiency.

**(15 Marks)**

(b) Describe, using examples, the main roles and their function in both XML and RESTful **Service Oriented Architectures** (SOA).

**(6 Marks)**

(c) Explain any advantages that a **SOA** may have over lower-level procedural and method-oriented remote communication models.

**(4 Marks)**

**3.** *"C.J. Date's rules for distributed databases define a set of objectives that must be satisfied by a fully Distributed Database System (DDBS)."*

(a) Discuss the degree to which **relational**, **wide-column** and **graph** database models can be considered to be fully DDBS using the criteria specified by C.J. Date.

**(15 Marks)**

(b) Explain how a distributed tuple store can exploit a **hash-ring** to promote both high availability and scalability. Include in your answer a diagram showing how a hash ring can be used to partition and locate database nodes.

**(10 Marks)**

**4.** (a) Describe, using UML and diagrams where relevant, the function of the following components of the **RMI architecture**:

- Remote Interfaces **(3 Marks)**
- Remote and Dynamic Proxies **(3 Marks)**
- Naming Services **(3 Marks)**

(b) Discuss how RMI can be used to simulate a **pass-by-reference**. Your answer should be accompanied by a diagram illustrating the component parts involved in the process and code snippets showing the key interactions.

**(11 Marks)**

(c) Briefly, explain how the RMI architecture can be adapted to enable direct communication with a remote **CORBA** orb.

**(5 Marks)**

**5.** (a) Using a fully labelled diagram, describe the major components of an **Object Request Broker** (ORB) and their function in the CORBA architecture.

**(13 Marks)**

(b) The following two Java interfaces abstract an *PurchaseOrder* and an *LineItem* respectively.

```
package ie.gmit.sw;
public interface LineItem {
  String itemNumber ();
  String name ();
  int quantity ();
  double price ();
}

package ie.gmit.sw;
public interface PurchaseOrder{
  String poNumber();
  void add (LineItem i);
  void remove (LineItem i) throws ItemNotFoundException;
  int count ();
  LineItem[] items ();
}
```

Translate the two Java interfaces into their CORBA **Interface Definition Language** (IDL) representation and show how the IDL module may be compiled and orchestrated into a set of server-side classes.

**(12 Marks)**