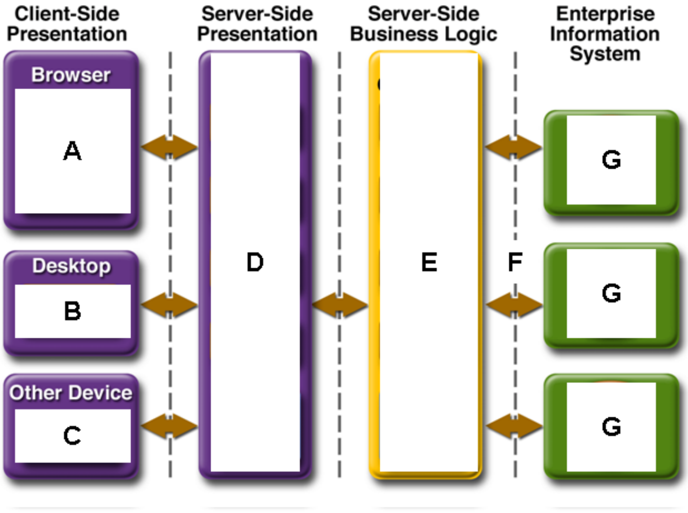
Samples questions for Software Engineering

J2EE

1. The following diagram describes the J2EE architecture.

a) Complete with the following sentences with the letter(s) in the figure so that they became true.



Servlet are in area(s)s …

Glassfish includes …

EJB are in area(s)…

JSF involves area(s)…

EntityManager is in area(s)…

Container are in area(s)…

JSP pages are in area(s)…

Derby can be in area(s)…

ManagedBean are in area(s)…

Webservice are in area(s)…

2. Give an example in J2EE architecture for the following patterns

Facade

Singleton

Model View Controller

Testing

3. Suppose that you are asked to implement a small application *TODOList* that manages a simple Todo list where each entry has subject and a date. You can add, edit and remove entries from the list. Of course you are expected to identify all the entries within a certain day range to be displayed and remove all those with more than 7 days after completion. The application should be use a MySQL database that is already installed.

Complete the vacant space either by defining a possible implementation using junit (omit annotations but) or completing the story

Scenario: adding an entry to the TODOList

Story 1: adding an valid entry to an empty TodoList	
Given an empty TodoList	// Given
And an Entry with today’s date and subject “hello”	TodoList todo=new TodoList();
When adding the Entry to the TodoList	Entry entry=new Entry(new Date(), “hello”);
Then TodoList has one entry	// When
And that entry subject is “hello”	todo.add(entry);
	//Then
	AssertEquals(todo.size(), 1);
	AssertEquals(todo.top().getSubject() , “hello”);

Story 2: adding an invalid entry to the TodoList	
Given	TodoList todo=new TodoList();
	Int cnt = todo.size();
	Date d=new Date();
When	d.setYear(2011);
	Entry entry=new Entry(d, “hello”);
	todo.add(entry);
Then	AssertEquals(todo.size(), cnt);

Story 3: adding an entry with no date to a TodoList	
Given a TodoList	
And an Entry without date and subject “hello”	
When adding the Entry to the TodoList	
Then TodoList has one entry	
And the last Entry has the date of today	

Junit and JMock

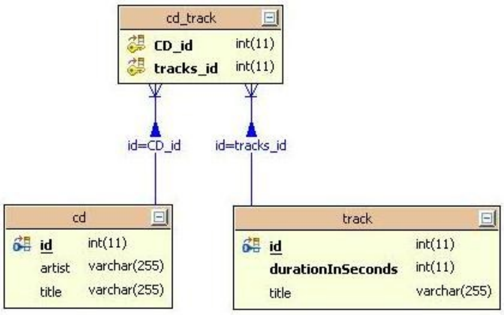
4. JUnit and Jmock: The following code is inserted in a junit test class (not complete)

```
private UserService userService = new UserServiceImpl();
private Mock userDAO = null;

protected void setUp() throws Exception {
  userDAO = new Mock(UserDAO.class);
  userService.setUserDAO((UserDAO)userDAO.proxy());
}
public void testGetUser() {
  User fakeUser = new User("John");
  userDAO.expects(once()).method("getUser").with(eq(1L)).will(returnValue(fakeUser));

  User user = userService.getUser(1L);
  assertNotNull(user);
  assertEquals("John", user.getName());
}
```

a) Explain **succinctly** what is happening in this test code. Suggestion: number the lines of code you think are relevant to support your answer and explain them.

JPA

5. Based on the following diagram (Entity-Relationship model), that describes that a CD has many tracks write a stripped java code that implements this schema using EJB entities. Include the fields and annotations (discard sizes and getters/setters).



Applied J2EE

6. Suppose you have been requested to assist the Mitwos specification. Propose a proper architecture to address the previous technical issues and explain how the expected features are addressed by your solution. The Mitwos specification follows:

The Milky Way Touristic Association (MWTA) is launching a new touristic programme with its associates. Among other features, the new business opportunity will enable:

- citizens can by a MilkyVoucher from one of the MWTA offices, to themselves or to offer; this voucher that can later be redeem at a given Hotel in the MWTA network.
- staff at the MWTA front offices, distributed along the region, will be able to check the availability of Hotel rooms in real time and make reservations (for the associated Hotels);
- MWTA customers will be able to make Hotel reservations through a friendly web portal.

The technical team in charge of implementing the underlying information system, the Milky Integrated Information System (Mitwos), has come up with the following technical requirements:

1. citizens will have access to a friendly web portal, providing a login, with a fancy design;
2. sellers/clerks at the MWTA offices will use a desktop application (check for availability, place reservations, run statistics, etc.);
3. Hotel's clerks will access a simple portal to check a MilkyVoucher status and mark it as used;
4. data is to be stored on an a PostgreSql relational database server, but, possibly, later changed to a better commercial supported relational database.
5. programming language and development paradigms must build on existing Object-oriented skills;
6. a strong transactional support is required in Mitwos, to ensure proper reservations placement;
7. Hotels in the MWTA network already use information systems; they should expose proper services so the Mitwos can query them for availability and place rooms reservations;
8. to ensure the better availability possibility, it should be possible to deploy different software pieces/services to different machines;
9. free software (both for development and deployment) is preferable.