Cairo University **Faculty of Computers and Artificial Intelligence**
Computer Science Department – Software Engineering Undergrad Program

SCS 491: Selected Topics in Software Engineering-1, MIDTERM EXAMINATION

Instructor: Dr. Soha Makady

Winter 2022 – CLOSED Book Exam –Total marks: 20 – Duration: 60 mins

This exam comes in **four** pages.

STUDENT NAME _____

STUDENT ID # _____

Note: Some questions demand explanations and/or justification. If no explanation and/or justification is provided, that question will receive a zero grade.

Question 1 [4 marks]

a) Define distributed systems. **[1]**

A distributed system is a set of heterogenous devices connected via a network to offer a number of shared services where this system appears to the user as a single entity with no knowledge about its internal details.

b) What is meant by heterogeneity in distributed systems? How can this challenge be solved in modern distributed systems? **[1]**

A heterogeneous distributed system supports different hardware architectures, operating systems, software interfaces, programming languages, network architectures, etc…

This challenge can be resolved in a single distributed system by providing unified middleware installed on all devices that comprise the underlying distributed system.

c) There is **no global state** in a distributed system. What is meant by this sentence? **[1]**

For any distributed system that involves main processes, it is not possible to know the current global state of the system. For instance, if you have a banking system with several branches, there is no chance that you know at some current time, the state of all bank account users, because several transactions can be happening at different processes at such moment.

d) When creating a **Java Socket**, a port number is required. What is the role of the port number in this process? **[1]**

The port number specifies a specific unique application that you need to access on a remote device.

Question 2 [6 marks]

(a) Suppose that a server (program) runs on a specific computer. Consequently, it will create a socket that is bound to a specific port number, e.g. 50. The server listens to the socket for a client to make a connection request. If everything goes well, the server accepts the connection. Upon acceptance, the server creates a new socket that is bound to a different port, e.g.., 1200. Why does a different port number, 1200, is required in this scenario other than the original port number, 50? **[2]**

The specific port number, i.e. 1200, is created on the server for a specific client after accepting its connection request is important to free the original port number, i.e. 50, for any upcoming connection requests from other clients. If the original port number remains the only port number that is used all the time with a single client, other client requests will be rejected.

(b) Draw **a diagram (or write an equivalent explanation)** to illustrate two possible ways to create threads in Java. Describe **the differences** between the two ways. **[2]**

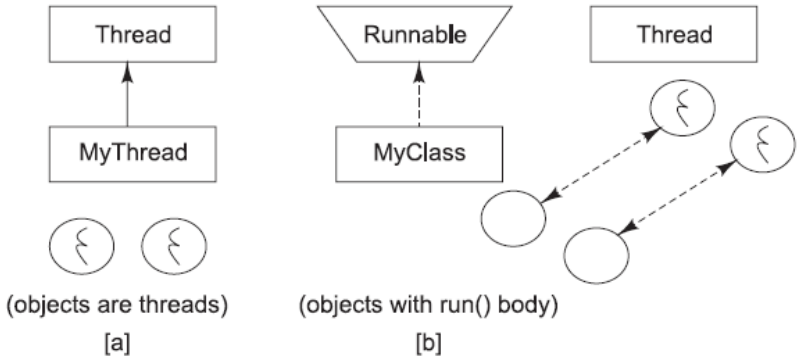


Fig. 14.3 Creation of Threads in Java

The main differences between the two ways are as follows:

(i) By extending the Thread class the derived class itself is a thread object and it gains full control over the thread life cycle.

(ii) By Implementing the Runnable interface developers do not have any control over the thread itself, since the run() method here simply defines the unit of work that will be executed in a thread.

(iii) Extending the Thread class is possible if you don't need to inherit additional classes, whereas if the class already inherits another class, implementing the runnable interface would be the alternative option.

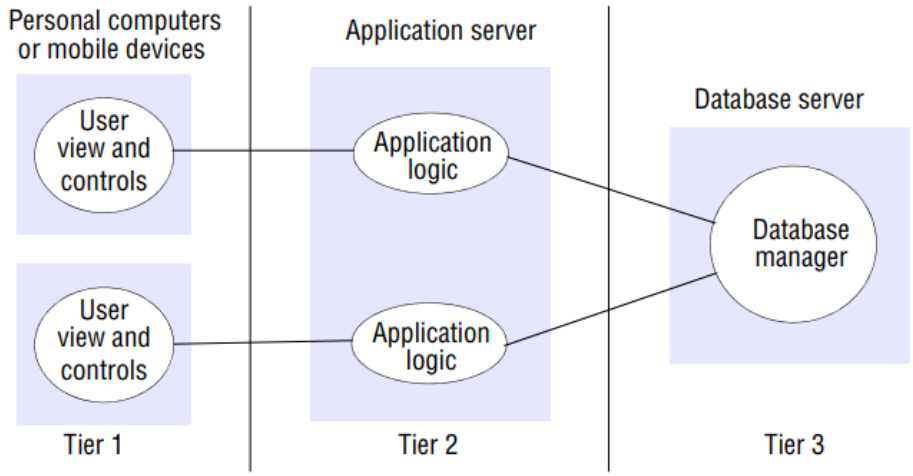
(c) To execute the following code, first t1 will run, second t2 will run, and finally t3 will run: **True** or **False** and **why? [2]**

```
Thread t1 = new Thread();
Thread t2 = new Thread();
Thread t3 = new Thread();
t1.start();
t2.start();
t3.start();
```

No guarantee for any specific sequence of running for the three threads since the Java virtual machine is fully responsible for thread scheduling irrespective of the sequence provided by the developer. This is correct unless the developer forces a specific sequence by assigning different priorities to the three threads.

Question 3 [4 marks]

Consider the below figure that illustrates a three-tiered architecture.



a) **Explain** one advantage of that architecture, and how that architecture enables such an advantage. **[1]**

Two possible answers: scalability **or** maintainability

Scalability because you can scale each tier separately by adding more machines (whether it is the application logic tier or the database tier.

Maintainability due to separation of concerns as each logical component is deployed to a separate tier, hence you can make changes to it, and re-deploy it without affecting the other tiers.

b) **Explain** one disadvantage of that architecture, and how that architecture leads to such a disadvantage. **[1]**

Latency because you would need through some external network connections twice (once across each pair of tiers). This is different from two tiers, where you would only need to go through one network connection level. Hence as you increase the tiers, the latency increases thus affecting the overall response time.

c) Consider some banking application. **Explain in detail** what would be present in "Tier 2" and 'Tier 3" for that application. Your explanation needs to be **specific to the banking application**. **[2]**

Tier 2 two would hold the application logic (e.g., validating that the customer has enough money to withdraw some amount x)

Tier 3 would hold the data logic (the database the holds all the information needed for the banking application like the customers' info and the accounts info)

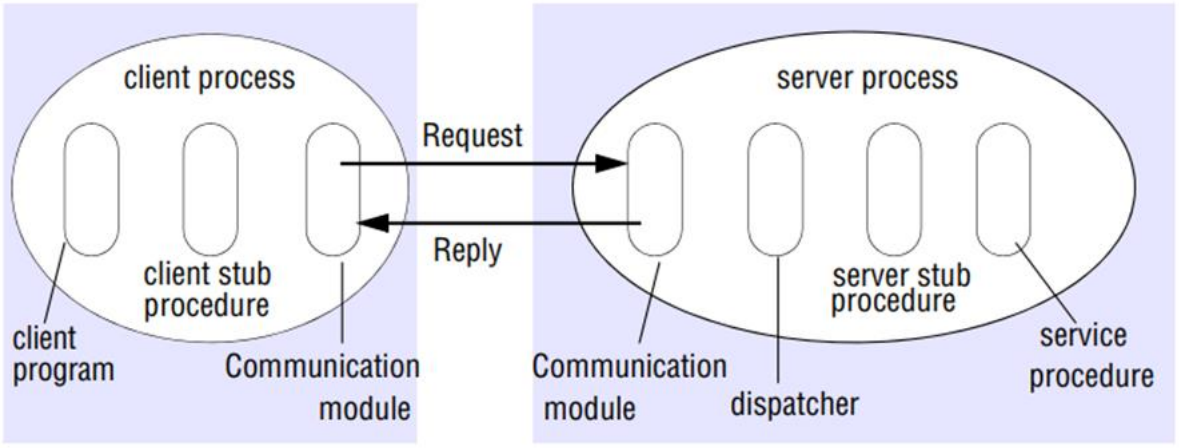
Question 4 [6 marks]

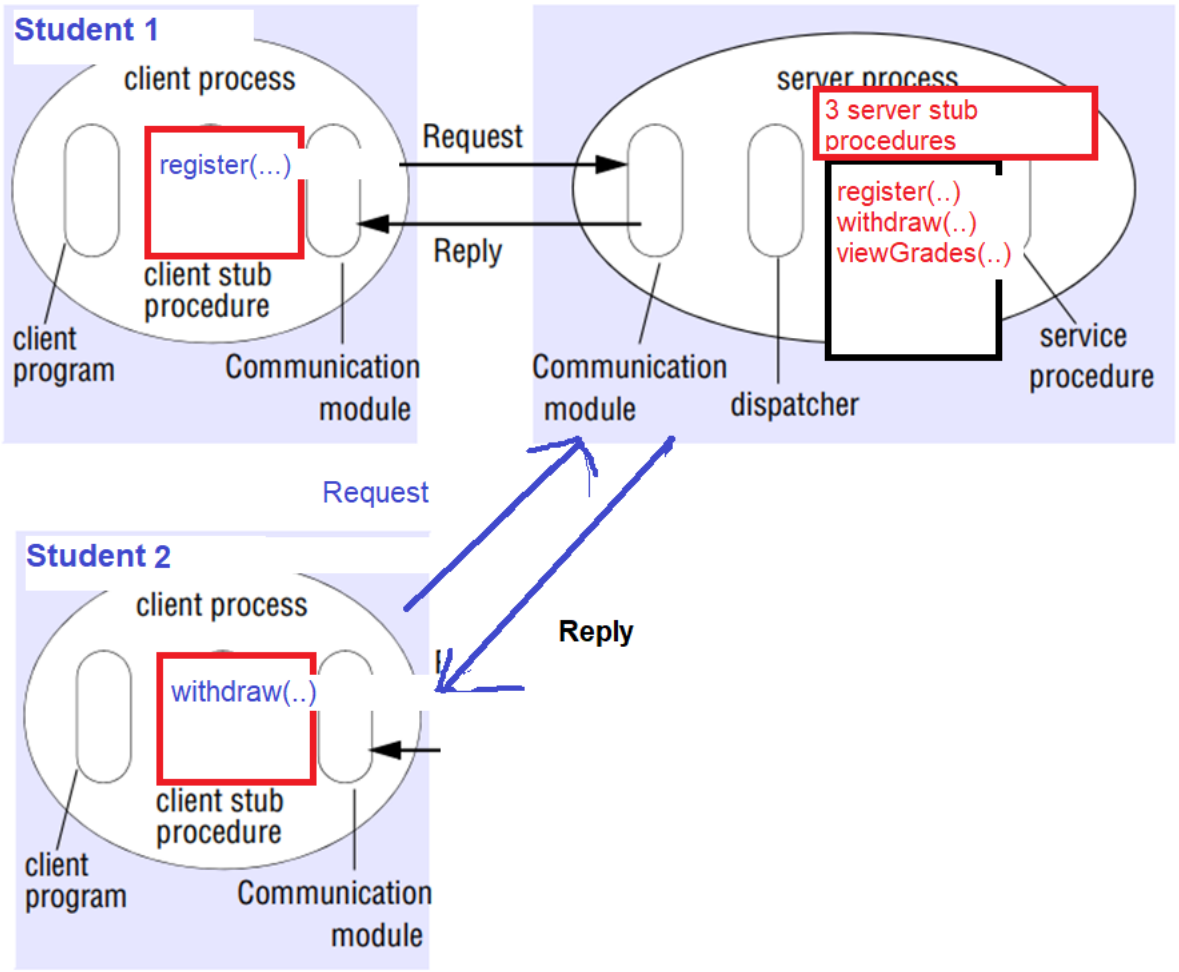
Consider the following diagram for the RPC architecture. Assume that you would like to utilize that architecture within an online university management system (like new e-com at our faculty). Such online management system would enable students to: **register** specific courses, **withdraw** from registered courses, and **view** their letter grades. Consider that you currently have two students: Student X who would like to register in a new course, and Student B who would like to withdraw from a course that he previously registered.

Given the above information about the online university management system, use the below figure to: (i) **sketch** a diagram illustrating the RPC communication within that system, and (ii) provide **written explanation**.

Your **diagram** should utilize the below figure notations and show: (i) the whole architecture involving the students and the server, (ii) the **content** of the "client stub procedure", (iii) the **content** of the "server stub procedure" for all communicating parties. Make any additional needed assumptions.

Your written explanation should clarify the complete flow of control (between the entities within the sketched diagram) since one student requests **to register a specific course** till that request gets executed on the server.





1. The student client program calls the client stub procedure for “register course” while passing the needed arguments.
2. The client stub procedure marshals the arguments, and the marshalled message along with the ID of the “register course” procedure.
3. The communication module sends the marshalled message to the server.
4. Based on the sent message, the dispatcher (on the server side) selects the server stub procedure for “register course”
5. The server stub procedure for “register course” unmarshalls the arguments.
6. The server stub procedure calls the service procedure for “register course” (i.e., the register course procedure is now executed on the server).

End of Exam