

University of Genoa
Polytechnic School - Master in Computer Engineering
Academic Year 2020-2021
Software Platform
Final Exam

Students are evaluated through an oral exam based on a personal session with the instructor. The oral exam requires the ability to develop programs as well as the ability to master the frameworks and the platforms, according to what was presented at the lectures.

Students must be able to interact with the program development tools, the frameworks and the platforms covered in the course both through a Command Line Interface and through a Graphical User Interface.

The candidate must bring his/her laptop with the following packages:

- Java jdk 8
- Eclipse IDE for Java EE Developers
- Apache Tomcat, Axis 2, Apache Ant, Apache Maven
- Eclipse Jersey
- Postman
- Oracle VirtualBox
- Linux Ubuntu on an Oracle VirtualBox appliance.
- Wireshark

These packages can be either in MS Windows (preferred) or in Linux, depending on the candidate preferences. In addition, the candidate must have the following packages running Linux Ubuntu:

- Tcpdump
- Apache Tomcat
- Docker

Finally, the candidate's laptop may contain the programs and the scripts presented and discussed in the course lectures, to be used as a reference. The candidate is invited to use the programs during the exam.

The course includes three sections, each in turn including four topics. The exam will focus on three topics, one for each section, drawn by the instructor. The candidate will be invited to present/discuss the topics using the programs/scripts presented at the lectures, and available on Aulaweb. The presentation/discussion of each topic will last approximately 15 mins and will require program/script/configuration analysis, execution and testing, network traffic observation, platform configuration and management, etc.

The sections/topics are listed below:

Section 1. Java technology in support of Software Platforms

- Topics 1.1: Socket and Asynchronous I/O
- Topics 1.2: Threads and thread pools
- Topics 1.3: Dynamic Class Loading, Reflection and Annotations
- Topics 1.4: Servlet container development

This first part of the exam requires the mastery of

- the aforementioned Java technologies
- the Windows/Linux CLI (depending on the candidate's Laptop configuration)
- the Eclipse IDE and of the appropriate Java Libraries.

For the students that have submitted the first homework this part of the exam will consist of a discussion of what they have submitted.

Section 2. Web Apps and Web Services

- Topics 2.1: Web Apps on Apache Tomcat servlet container
- Topics 2.2: Remote Method Invocation (RMI)

Topics 2.3: SOAP Web Services

Topics 2.4: REST Web Services

The second part of the exam requires the mastery of:

- a Web Apps framework (e.g., Apache Tomcat)
- a SOAP Web Service framework (e.g., Apache Axis 2)
- a REST Web Service framework (e.g., Eclipse Jersey)
- Wireshark, Eclipse and the CLI, on the client side, using Windows or Linux, depending on the candidate's Laptop configuration
- Tcpcat and all the other appropriate tools in the Linux OS on the server side through a CLI.

Section 3. Virtual Environment creation, management and monitoring

Topics 3.1: Virtual Machine Management and Configuration

Topics 3.2: Linux Namespaces technology for Container support

Topics 3.3: Docker Container basics

Topics 3.4: Web Apps, Web Services and Load Balancing on Docker Containers.

The third part of the exam requires the mastery of:

- Virtual Machine configuration (Ref. Oracle Vbox) both through the GUI and through the CLI
- Linux Namespaces
- Docker Containers.

More specifically, the sections/topics cover:

1. First Section: Basic Java Technology for Software Platforms

The candidate must be able to discuss the following topics through the analysis/development of Java programs:

- Socket
 - Client Server interaction over a Stream Channel (TCP);
 - Http encapsulation.
- Asynchronous I/O:
 - Asynchronous I/O applications based on the Selector class.
- Thread:
 - Implementation of computing-intensive application and waiting-intensive applications on a single thread and on a variable number of threads;
 - From Threads to Thread Pools (ExecutorService);
 - Thread Pool implementation (BlockingQueue).
- Dynamic Class Loading
 - Platform Management and run time Application Deployment
- Reflection:
 - Analysis of classes, methods, etc. to support application deployment on Software Platforms.
- Annotations
 - Annotations with different Retention Policies. In particular: RetentionPolicy.RUNTIME to support Software Platform configuration.
- Servlet Container
 - Development of a simple servlet container incorporating socket programming, thread-based programming and pools, dynamic class loading, reflection, annotations. Implementation of a CLI oriented management console as a separate thread. See the first Homework.

2. Second Section: Frameworks and Platforms to host Services and Applications

2.1. Frameworks and Platforms to host Services and Applications

- The Apache Tomcat Servlet Container
 - Download, Installation and Configuration of Tomcat.
 - Design, implementation and deployment of simple Web Application.
- SOAP Web Services
 - The Remote Method Invocation (RMI) paradigm for inter-process communication
 - From Remote Method Invocation to Web Services: WSDL and SOAP protocols
 - Analysis of the SOAP Web Services data flow through the Apache Axis 2 framework.
- REST Web Services
 - From Http to REST
 - Implementation of REST Web Services and Clients using the Eclipse Jersey Framework
 - Analysis of REST Web Services using Postman

2.2. Virtual Machine management with CLI

- Virtual Machines (Reference Example: Server Side Oracle Vbox (only CLI))
 - VM manager download, Linux appliance download, configuration, testing and analysis. Virtual Networking, Shared Clipboard and Directories (Ref. Oracle Vbox).
 - Installation of software tools and frameworks on Linux (tcpdump, net-tools, ip-utils-ping, Tomcat, Axis 2)
- Analysis of Web Apps and Services over the internal Virtual Network
 - Porting of server-side software to the Linux VM
 - Packet inspection and protocol analysis of client-server interactions using Wireshark on the client side and tcpdump, on the server side.

3. Third part: From Container technology to Docker

- Linux technologies to support containers.
 - Analysis of Linux namespaces technology.
 - Shell Scripts and C programs to create and manage segregated applications. In particular:
 - Direct Connection between namespaces.
 - Bridged Connection between two namespaces
 - Connection from a Network Namespace to the external world through routing/natting.
 - Program controlled namespace activation through the clone() function.
 - Program controlled container join through setns().
 - Program controlled activation of a Tomcat container and access from outside.
- Docker Containers
 - The Docker container support system;
 - Docker download, installation, configuration, provisioning;
 - Images, containers, networks, volumes;
 - Local and remote repository;
 - Image building (Dockerfile);
 - Implementation of Web Apps and of Web Services of distributed container architectures;
 - Load Balancing using Nginx and Docker.