University of Genoa
Polytechnic School - Master in Computer Engineering
Academic Year 2020-2021
Software Platform
Reference Material – Dec. 9, 2020

The course consists of the following three parts:

1. Basic Java Technology for Software Platforms
2. Frameworks and Platforms to host Service and Applications
3. Container Technology

The course reference documentation is organized as follows:

1. The reference documentation for the first part of the course includes:
   - The appropriate Java library documentation, and in particular
     - Socket
       https://docs.oracle.com/javase/8/docs/api/java/net/Socket.html
       https://docs.oracle.com/javase/8/docs/api/java/net/ServerSocket.html
       https://docs.oracle.com/javase/8/docs/api/java/io/OutputStream.html
       https://docs.oracle.com/javase/8/docs/api/java/io/PrintStream.html
       https://docs.oracle.com/javase/8/docs/api/java/io/InputStreamReader.html
       https://docs.oracle.com/javase/8/docs/api/java/io/BufferedReader.html
       https://docs.oracle.com/javase/8/docs/api/java/io/BufferedWriter.html
       https://docs.oracle.com/javase/8/docs/api/java/net/InetAddress.html
     - Threads
       https://docs.oracle.com/javase/8/docs/api/java/lang/Runtime.html
       https://docs.oracle.com/javase/8/docs/api/java/lang/Thread.html
       https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/ExecutorService.html
       https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/BlockingQueue.html
     - Asynchronous I/O
       https://developer.ibm.com/tutorials/j-nio/
       https://docs.oracle.com/javase/8/docs/technotes/guides/io/index.html
       https://docs.oracle.com/javase/8/docs/api/java/nio/channels/SocketChannel.html
       https://docs.oracle.com/javase/8/docs/api/java/nio/channels/Selector.html
       https://www.tutorialspoint.com/java_nio/java_nio_selector.htm
       https://docs.oracle.com/javase/8/docs/api/java/util/Hashtable.html

- Dynamic Class Loading
  https://docs.oracle.com/javase/8/docs/api/java/lang/Class.html
  https://docs.oracle.com/javase/8/docs/api/java/lang/ClassLoader.html
  https://docs.oracle.com/javase/8/docs/api/java/net/URL.html
  https://docs.oracle.com/javase/8/docs/api/java/net/URLClassLoader.html
- Reflection
  https://docs.oracle.com/javase/tutorial/reflect/index.html
- Annotations
  https://docs.oracle.com/javase/tutorial/java/annotations/
- Servlets
  https://jcp.org/en/introduction/overview
  https://jcp.org/aboutJava/communityprocess/final/jsr340/index.html
  https://javaee.github.io/javaee-spec/javadocs/javax/servlet/http/HttpServlet.html
  https://javaee.github.io/javaee-spec/javadocs/javax/servlet/http/HttpServletRequest.html
  https://javaee.github.io/javaee-spec/javadocs/javax/servlet/http/HttpServletResponse.html

- The programs presented at the lectures, developed by the instructor to support the analysis of the technologies covered. In particular:
  - Socket and Asynchronous I/O
    A number of socket-based client-server programs plus a textual Description file.
    Program implementing an asynchronous client interacting with a synchronous server
    Program implementing an asynchronous client container
    Program implementing an asynchronous server container
  - Threads
    Program to retrieve the number of CPUs
    Program to measure CPU speed in a computing intensive application
    Program to measure the performance gains of multithreaded applications taking advantage of hardware parallelism
    Program to measure thread concurrency (overlapping processing and I/O)
    Program to test thread pools
    Program to implement thread pools through a blocking queue.
    Program to associate socket connections to threads
  - Dynamic Class Loading, Annotations and Reflection
    Program showing implicit vs explicit dynamic class loading
    Programs showing a simple example of a platform, operated by platform manager, and of an application, developed and deployed by an application developer, sharing the interface.

Program showing the way run-rime annotations work
Program that extracts the internal structure of a class through reflection.

- Servlets
  A simple servlet container based on socket programming, thread-based programming and thread pools, dynamic class loading, reflection, annotations.

2. The reference material for the second part of the course includes:
   - The documentation of the frameworks and of the technologies presented
     - The Apache Tomcat documentation
       http://tomcat.apache.org/
     - Remote Method Invocation
       https://docs.oracle.com/javase/tutorial/rmi/index.html
       https://docs.oracle.com/javase/7/docs/technotes/tools/solaris/rmiregistry.html
       https://docs.oracle.com/javase/7/docs/technotes/guides/rmi/hello/hello-world.html#define%60
     - WSDL/SOAP Web Services
       https://www.ibm.com/developerworks/webservices/tutorials/ws-understand-web-services1/ws-understand-web-services1.html
       https://docs.oracle.com/javaee/7/tutorial/webservices-intro.htm
       https://docs.oracle.com/javase/8/docs/api/java/net/URL.html
       https://docs.oracle.com/javase/8/docs/api/javax/xml/namespace/QName.html
       https://docs.oracle.com/javase/8/docs/api/javax/xml/ws/Service.html
       https://docs.oracle.com/javase/8/docs/api/javax/xml/ws/Endpoint.html
       https://www.javaworld.com/article/3215966/java-language/web-services-in-java-se-part-2-creating-soap-web-services.html
       https://docs.oracle.com/javase/7/docs/technotes/tools/share/wsimport.html
       https://www.w3.org/TR/soap/
       https://www.w3.org/TR/wsdl.html
     - The Apache Ant Documentation
       https://ant.apache.org/
     - The Apache Axis 2 documentation
       http://axis.apache.org/axis2/java/core/
     - REST Web Services
       https://docs.oracle.com/javaee/7/tutorial/webservices-intro.htm
       https://docs.oracle.com/javaee/7/tutorial/jaxrs.htm
       https://download.oracle.com/otn-pub/jcp/jaxrs-2_1-final-eval-spec/jaxrs-2_1-final-spec.pdf
       https://docs.oracle.com/cd/E24329_01/web.1211/e24983/overview.htm#RESTF105
       https://docs.oracle.com/javaee/7/tutorial/jaxrs002.htm

https://docs.oracle.com/javaee/7/tutorial/jaxrs-client003.htm
    - The Eclipse Jersey Documentation
      https://eclipse-ee4j.github.io/jersey/
    - The JSON documentation
      http://www.json.org/
    - The Postman documentation
    - https://www.getpostman.com The Google GSON documentation
      https://github.com/google/gson/blob/master/UserGuide.md
    - The Oracle Virtual Box documentation
      https://www.virtualbox.org/
- The programs and configurations presented at the lectures, developed by the instructor to support the analysis of the technologies covered. In particular:
    - Simple Web Applications based on GET and POST
    - Simple RMI based applications
    - Some simple SOAP Web Services and clients based on Apache Tomcat + Apache Axis 2
    - Some simple REST Web Services and clients based on Eclipse Jersey
    - A Simple Gson example

3. The reference material for the third part of the course includes:
    - The documentation on Linux namespaces technology and in particular:
      https://lwn.net/Articles/531114/
      https://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-rg-en-4/
      https://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-rg-en-4/ch-iptables.html
      https://man7.org/linux/man-pages/man2/clone.2.html
      https://man7.org/linux/man-pages/man2/setns.2.html
    - The programs and configurations presented at the lectures, developed by the instructor to support the analysis of the technologies and platforms covered. In particular:
        - Direct Connection between namespaces
        - Bridged Connection between 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.
    - The Docker documentation

https://www.docker.com/
https://docs.docker.com/engine/reference/commandline/cli/
https://hub.docker.com/
http://nginx.org/en/docs/http/load_balancing.html

- The programs and configuration presented at the lectures, developed by the instructor to support the analysis of the technologies and platforms covered. In particular:
  - Basic Docker entities (image, container, network, and volume) and commands to manage such entities (create, run, rm, ls, inspect, etc.)
  - Creating Images in the local repository through commit. Pushing/pulling images in the Docker repository.
  - Building Docker Containers: e.g., from scratch and from Busybox to show the Dockerfile structure.
  - Creating Linux Docker Containers, Tomcat Docker Containers, sharing webapps directory.
  - Load Balancing on Docker (Ref. nginx server).