Lab 1

1. Write the following client-server applications using the classes Server and ServerSocket from Java
   (a) the server component simulates the echo service of the operating system that reply with the string that the client component sends;
   (b) the server component simulates the NTP service that reply with the information related to the current time;
2. Write a client component that inspects the openess of the ports of the operating system of a certain computer (port number < 1024).

Lab 2

1. Write the following client-server applications using the classes DatagramPacket and DatagramSocket from Java
   (a) the server component simulates the echo service of the operating system that reply with the string that the client component sends;
   (b) the server component simulates the NTP service that reply with the information related to the current time;
2. Write the following application using the classes MulticastSocket from Java
   (a) a listener component that reads the message another messenger component sends to the group (launch multiple listeners)
   (b) a chat client component with a graphical interface that uses the group communication and launch multiple clients.
3. Write an application that is using remote method invocation in Java:
   (a) an interface that is common to the server and the client presenting the method that is remotely invocated by the client and implemented by the server
   (b) the method returns a string with a greeting message;
   (c) the server implements the method and is registering its method to the rmiregistry
   (d) the rmic is used as compiler to generate the stub for transforming the remote method call into a message exchange
   (e) the client calls the method and print the greeting message.

Lab 3

1. Write a web service in a bottom-up approach:
   (a) ensure that a Dynamic Web project can be created (e.g. WTP package is available in Eclipse), as well as an Tomcat installation is connected with the programming environment;
   (b) write in a Dynamic Web Project a class that has a method able to operate on two data (e.g. adding operation);
   (c) register the method/class as web service available in a Tomcat container.
2. Write an client service that tests the web service using the automation facilities of the programming environment (e.g. in Eclipse).
3. Repeat the aboves with a diversified list of methods that are publicly exposed.

Lab 4

1. Write a client component that call the method exposed as Web service from the Lab 3 and returns the result in console.
2. Write a client component that access two public web services

Lab 5

1. Write a friedly servlet (printing a “hello world” kind of message) that is active

   (a) either in azurewebsites.net

   (b) either in the local instance of Google App Engine and on appspot.com

   Follow the installation instructions from Azure, respectively Ggogle App Engine and the simple examples that they provide.

2. Change the previous application to display the time of at the server where is running.

Lab 6

1. Create an account on Amazon Wen Services Free Tier (https://aws.amazon.com/free/)
2. Create and deploy a Lambda function which outputs a greeeting message
3. Create and deploy a Lambda function which displays the time of at the server where is running.

Lab 7

1. Infrastructure-as-a-Service simulator with CloudSim
   (a) what is CloudSim: read here
   (b) how to install: read here
   (c) how to use: read here
   (d) follow the 8 examples from the CloudSim example directory
   (e) build your own simulation using 2 DataCenters, 8 Virtual Machines and 10 Cloudlets
2. Fog simulator with iFogSim
   (a) what is iFogSim and where can be found: read here
   (b) how to install: read here
   (c) follow the GUI usage examples and the simulation of VRGame
   (d) modify the simulation of VRGame to include more devices and users