University of Westminster
Faculty of Science and Technology

Module code and title: 5COSC004W-Client Service Architecture Tutorial Manual	
Tutorial title	**Getting acquainted with concepts and tools**
Tutorial type	**Guided and indepenent and non-marked**
Week 01	**23/01/2020**

Contents

Learning Goals

This tutorial focuses on two main learning goals:

- to get acquainted with the (some of the) tools you will be using during the module, and,
- to review basic java programming skills,
- to self-assess the status of their basic java skills

It is divided into two separate sections, the student will perform the first task (1-following the instructions of the tutor, and then, will complete the other tasks independently.

TASKS to be Performed under the instruction of the Tutor (from Task 1 to Task 10)

1) Start Netbeans in your system. If Netbeans is not present in your system, use AppsAnywhere to launch it: (https://www.westminster.ac.uk/sites/default/public-files/general-documents/Using%20AppsAnywhere.pdf)
2) Create a new java project in Netbeans (Figure 1 to 3)

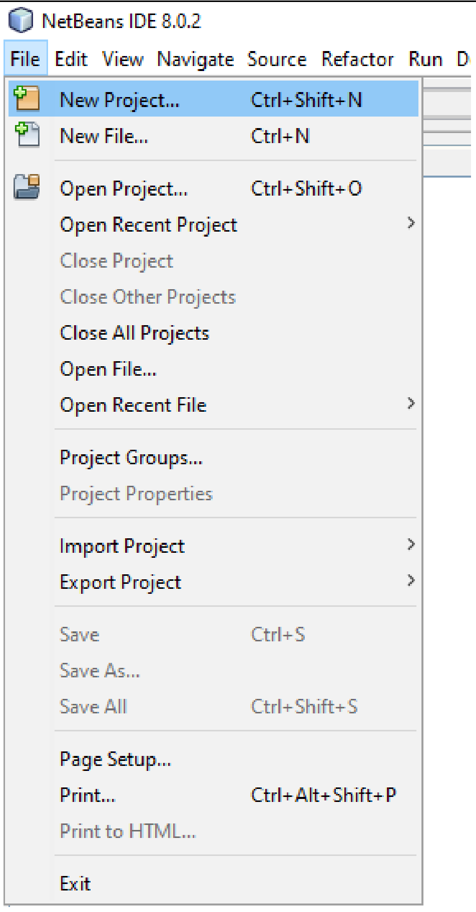
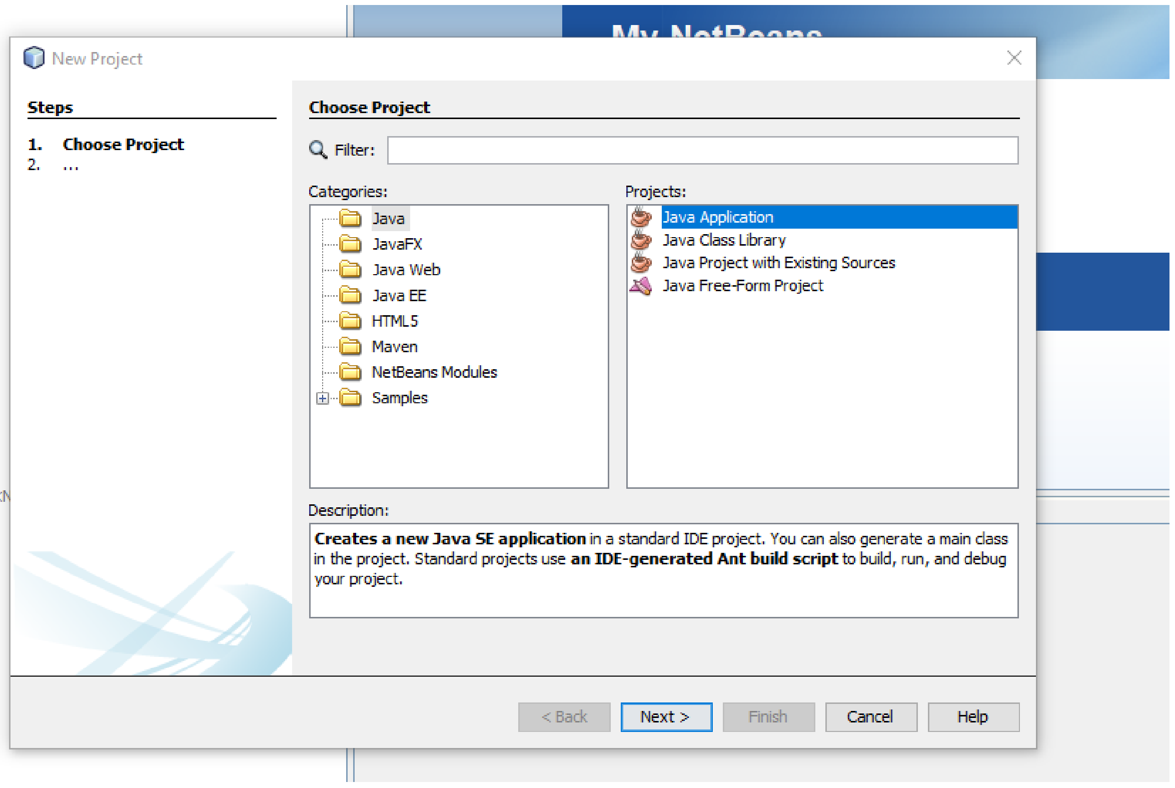
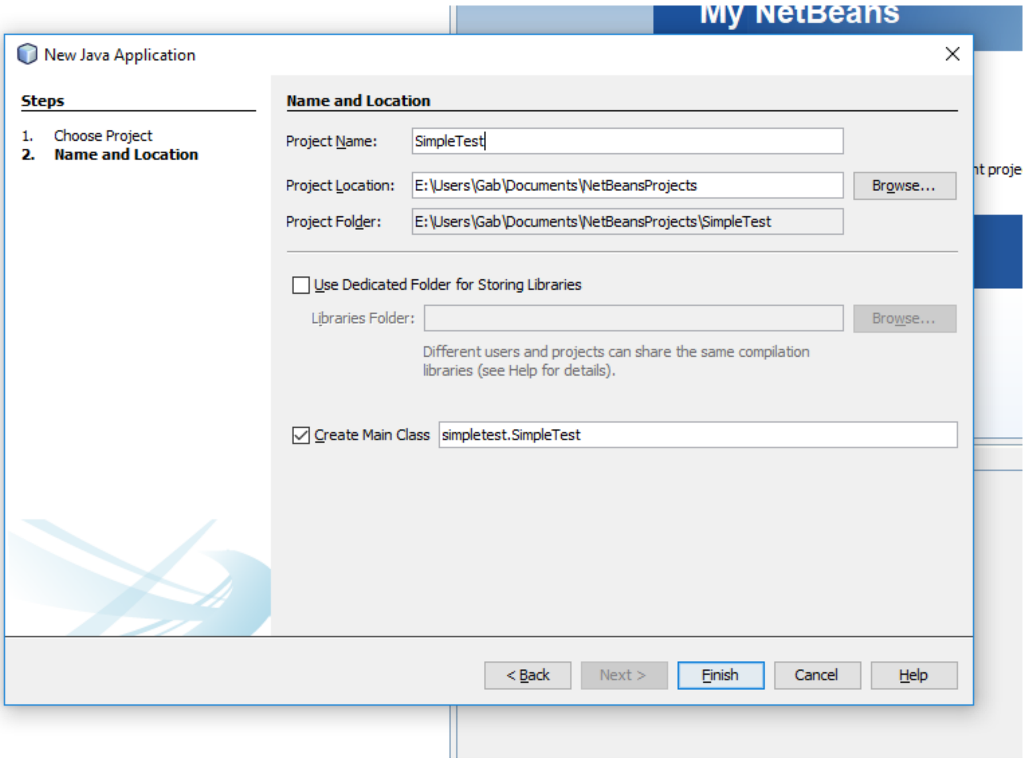


Figure 1, Create a New Project in NetBeans (1)



Creates a new Java SE application in a standard IDE project. You can also generate a main class in the project. Standard projects use **an IDE-generated Ant build script** to build, run, and debug your project.

Figure 2, Create a New Project in NetBeans (2)


New Java Application

Steps
1. Choose Project
2. **Name and Location**

Name and Location

Project Name: SimpleTest
Project Location: E:\Users\Gab\Documents\NetBeansProjects
Project Folder: E:\Users\Gab\Documents\NetBeansProjects\SimpleTest

☐ Use Dedicated Folder for Storing Libraries
Libraries Folder:
Different users and projects can share the same compilation libraries (see Help for details).

☑ Create Main Class simpletest.SimpleTest

Figure 3,Create a New Project in NetBeans (3)

3) Create a new java class in the project (Figure 1 to 3). The default Java Project created by Netbeans already has an empty SimpleTest with and empty constructor (Figure 4). We will use this simpletest empty class as the skeleton for our dummy client.

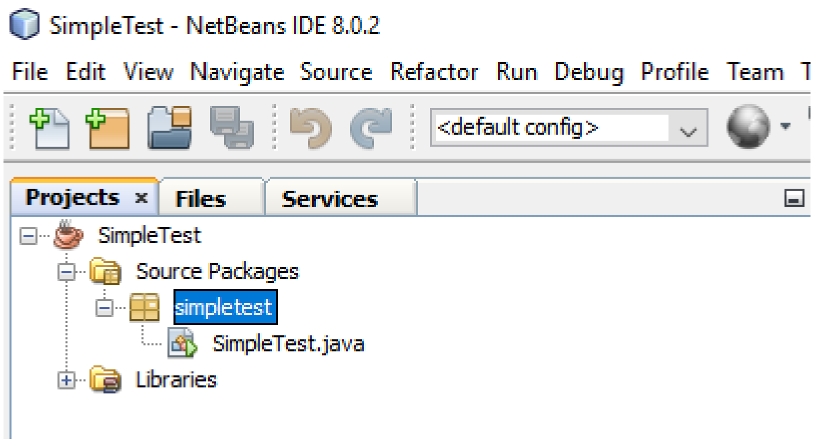

SimpleTest - NetBeans IDE 8.0.2

Figure 4, Standard Java Project in NetBeans

4) Create a new java class in the project (Figure 1 to 3) which will represent our Dummy Server.

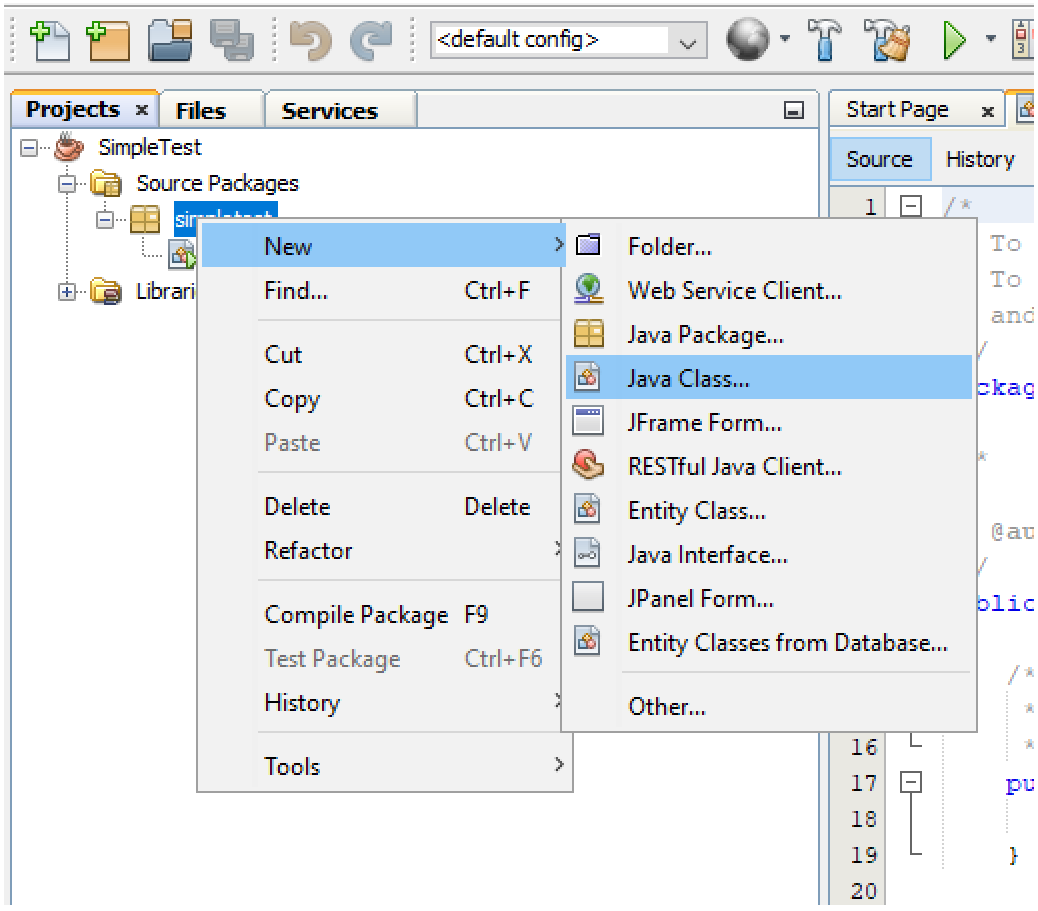
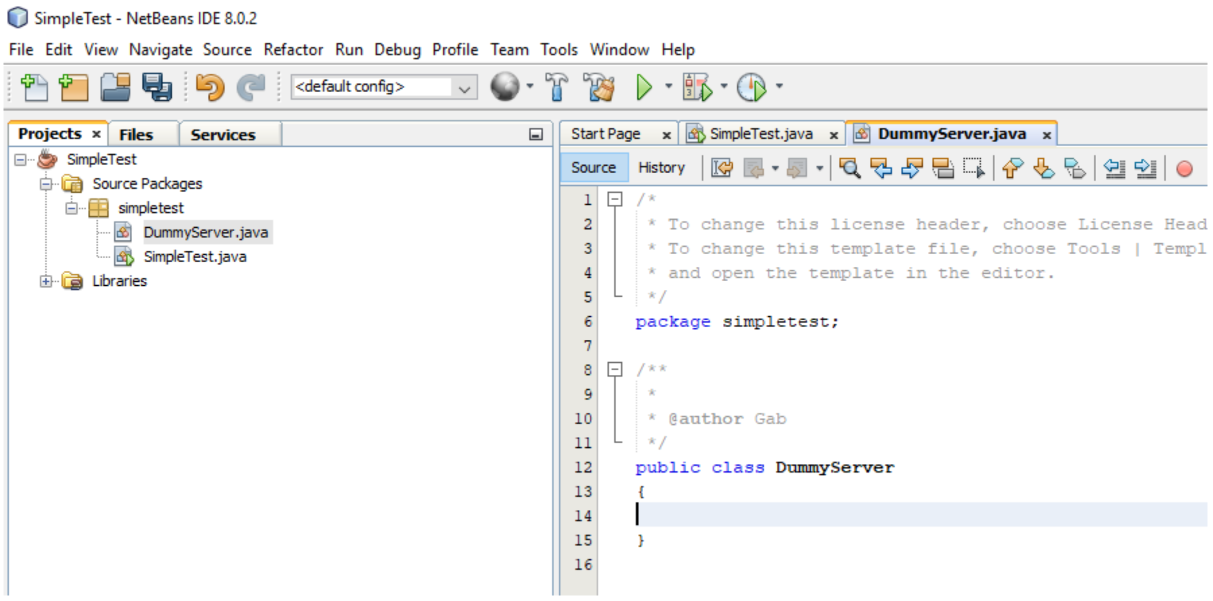


Figure 5, Create a new Java Class in NetBeans (1)



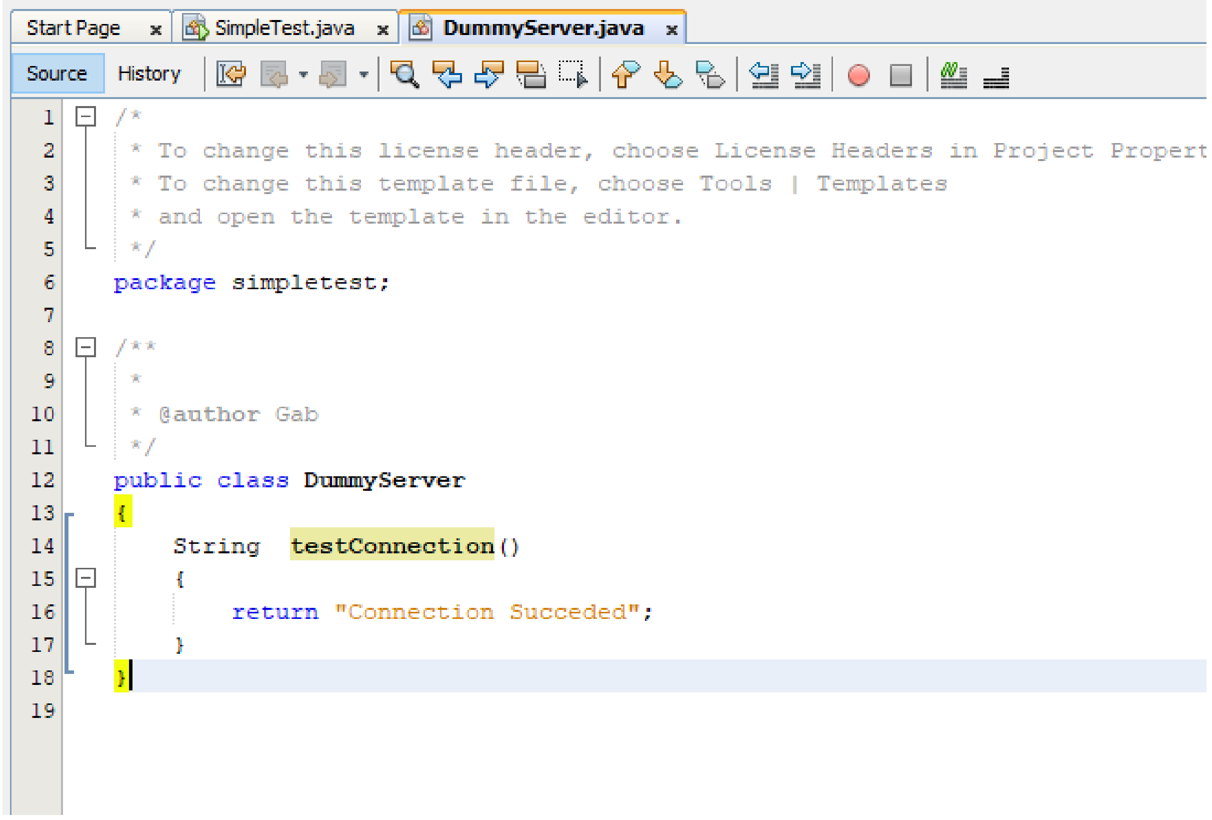
```
/*
 * To change this license header, choose License Head
 * To change this template file, choose Tools | Templ
 * and open the template in the editor.
 */
package simpletest;

/**
 *
 * @author Gab
 */
public class DummyServer
{

}
```

Figure 6, Create a new Java Class in NetBeans (2)

5) Add a method to the DummyServer class(Figure 6) that returns a simple string which we can test to see if the server is running.

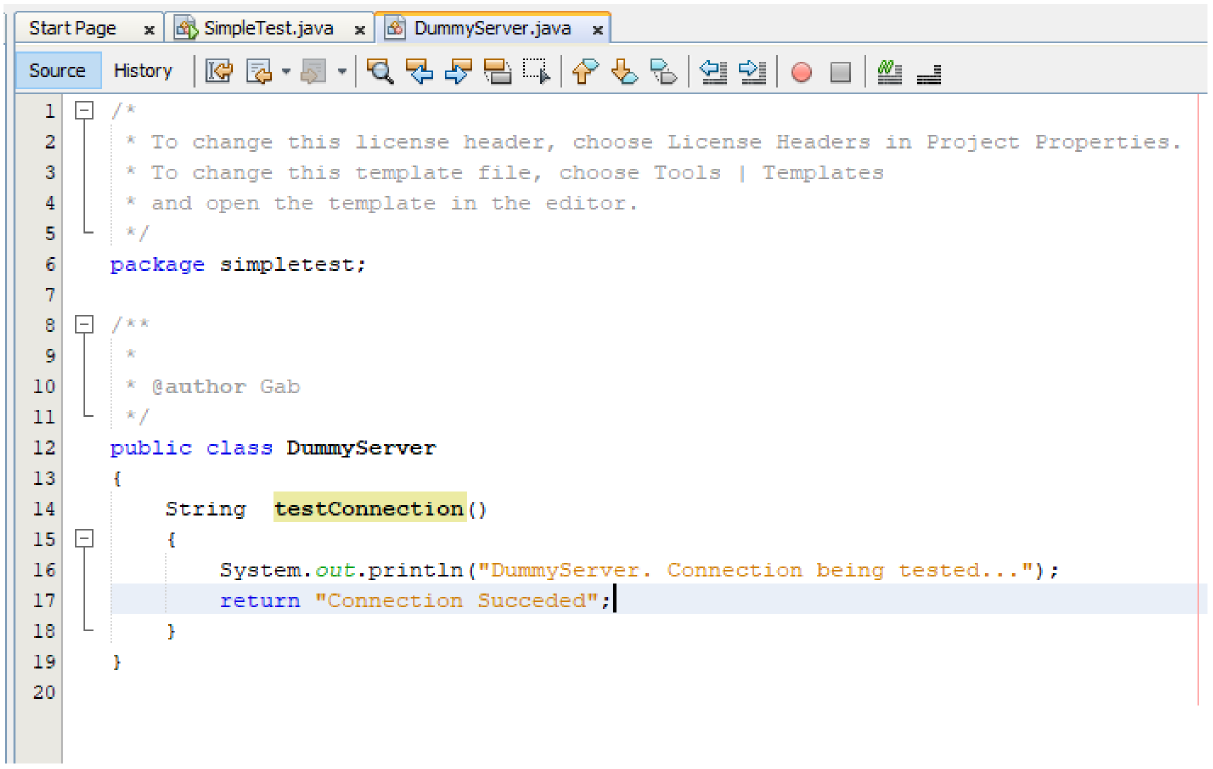


```java
/*
 * To change this license header, choose License Headers in Project Propert
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package simpletest;

/**
 *
 * @author Gab
 */
public class DummyServer
{
    String  testConnection()
    {
        return "Connection Succeded";
    }
}
```

Figure 7, Add simple method to the DummyServer Class

6) Add a log method to the DummyServer class that prints a simple string which we can test to see the debug the server while it is running.

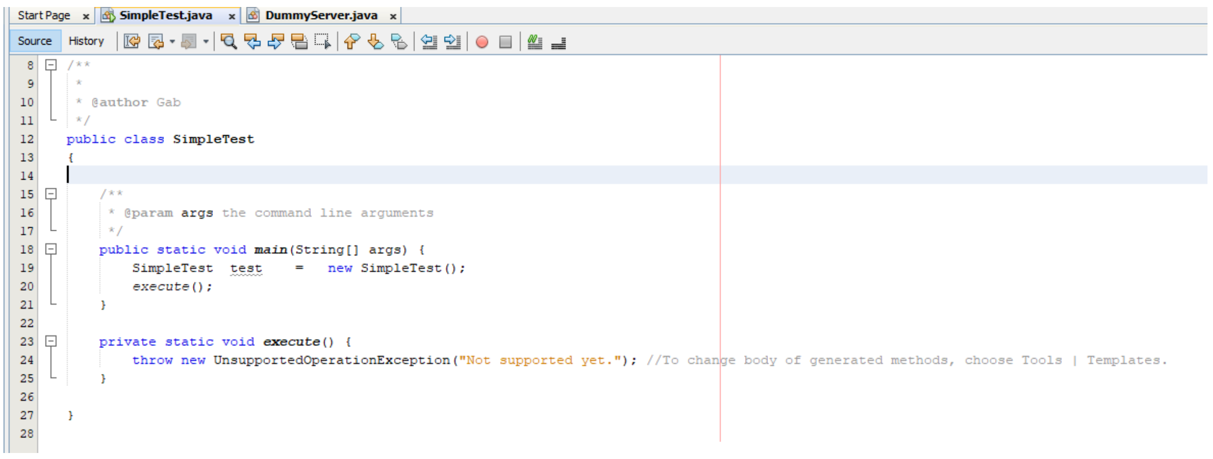


```java
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package simpletest;

/**
 *
 * @author Gab
 */
public class DummyServer
{
    String  testConnection()
    {
        System.out.println("DummyServer. Connection being tested...");
        return "Connection Succeded";
    }
}
```

Figure 8, Add simple logging to the method

7) Create an instance of the SimpleTest class and an empty method execute (Figure 9)



```
/**
 *
 * @author Gab
 */
public class SimpleTest
{

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        SimpleTest  test    =   new SimpleTest();
        execute();
    }

    private static void execute() {
        throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.
    }

}
```

Figure 9, Create instance and execute method.

8) Run the client (Figure 10-11)

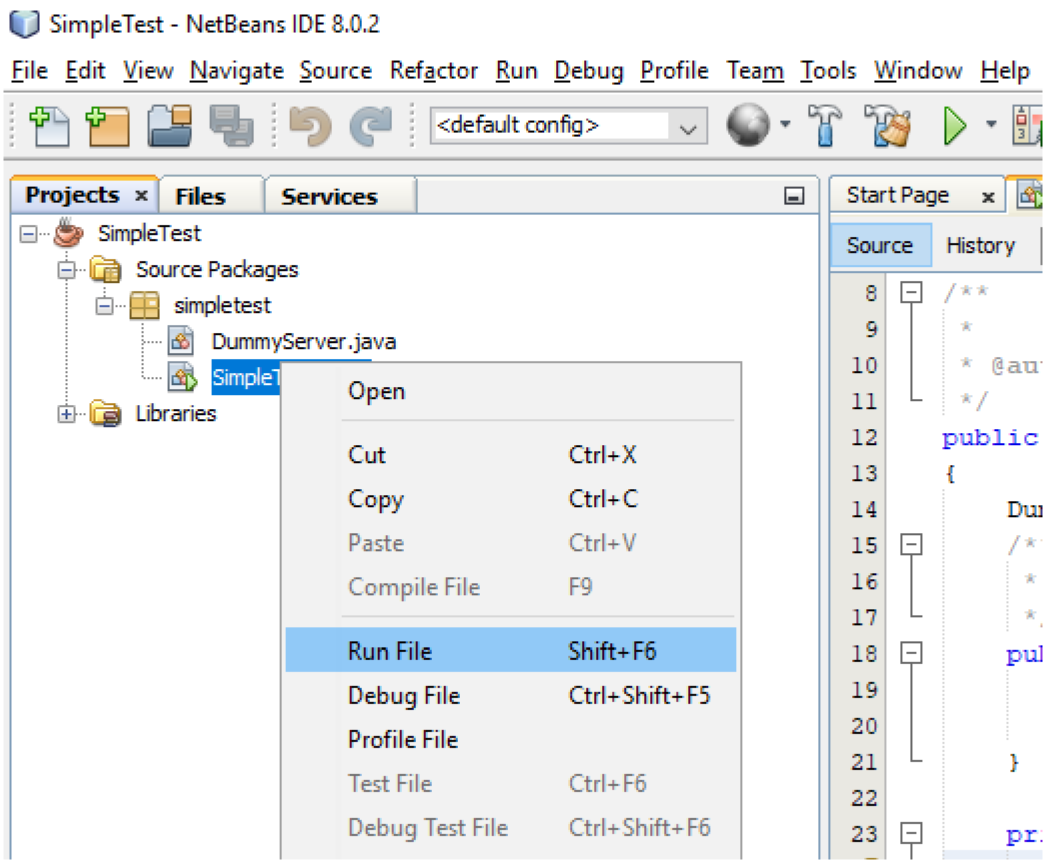
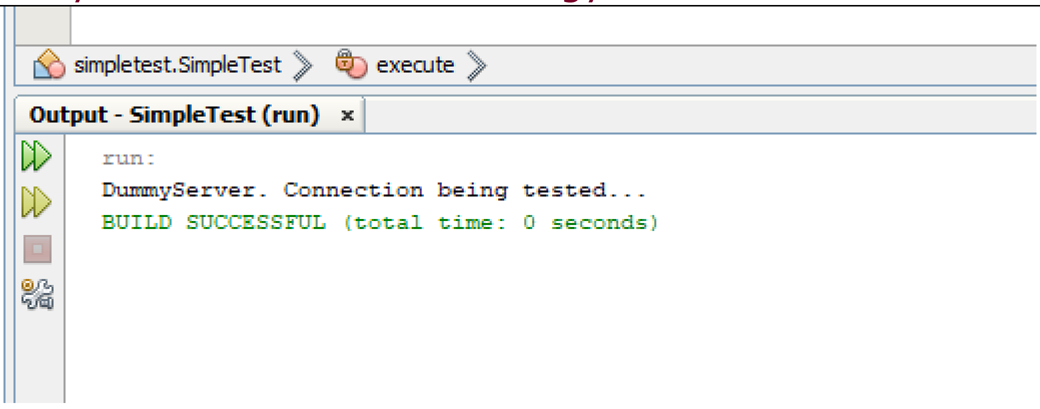


Figure 10, Run the test (1)



```
run:
DummyServer. Connection being tested...
BUILD SUCCESSFUL (total time: 0 seconds)
```

Figure 11, Run the test (2)

9) Export the project as zip file on Netbeans (Figure 12-13)

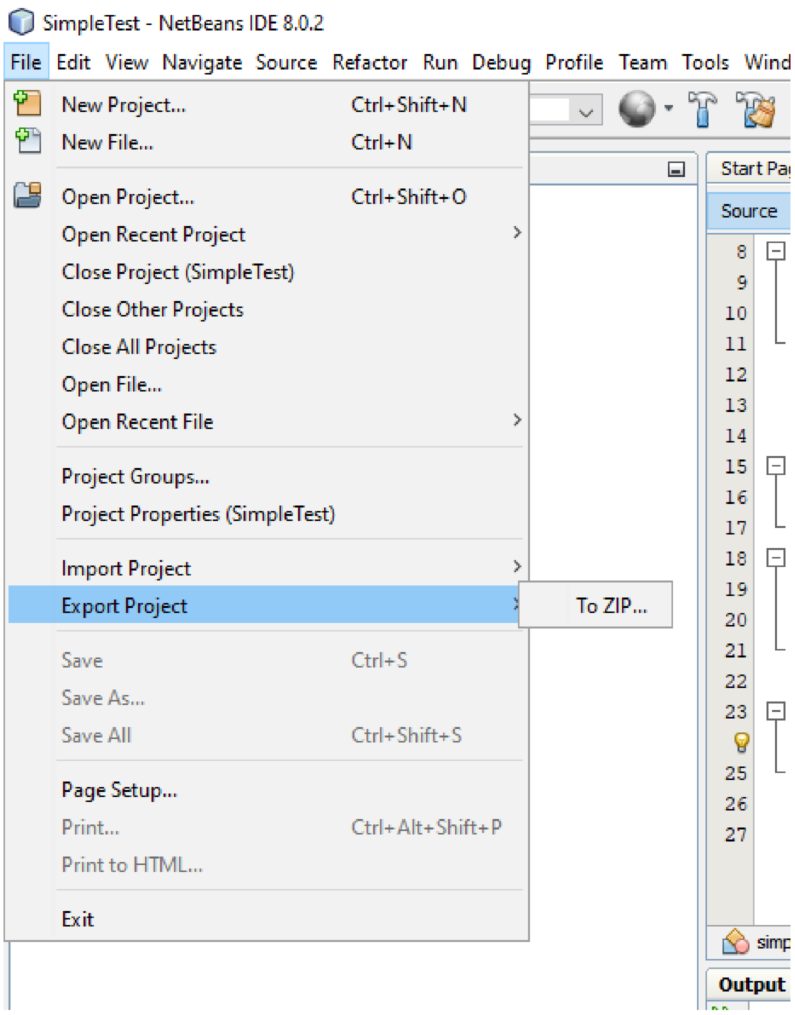
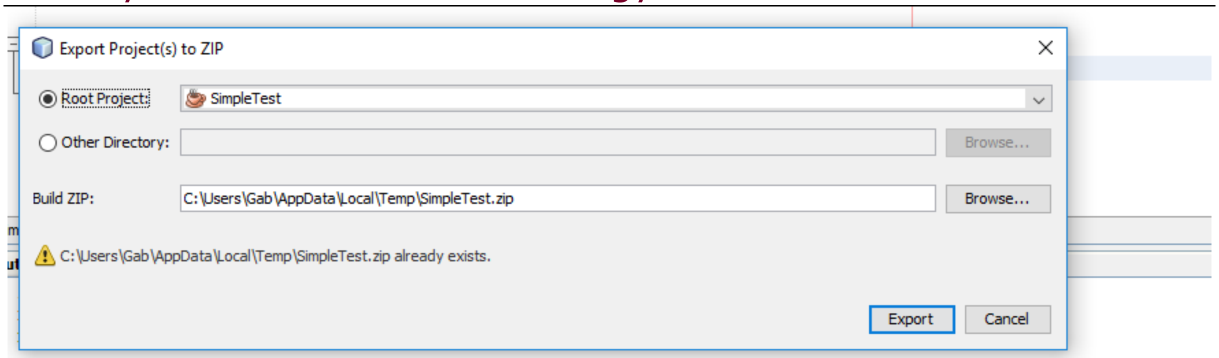


Figure 12, Export the project as zip file (1)


Export Project(s) to ZIP

Root Project: SimpleTest

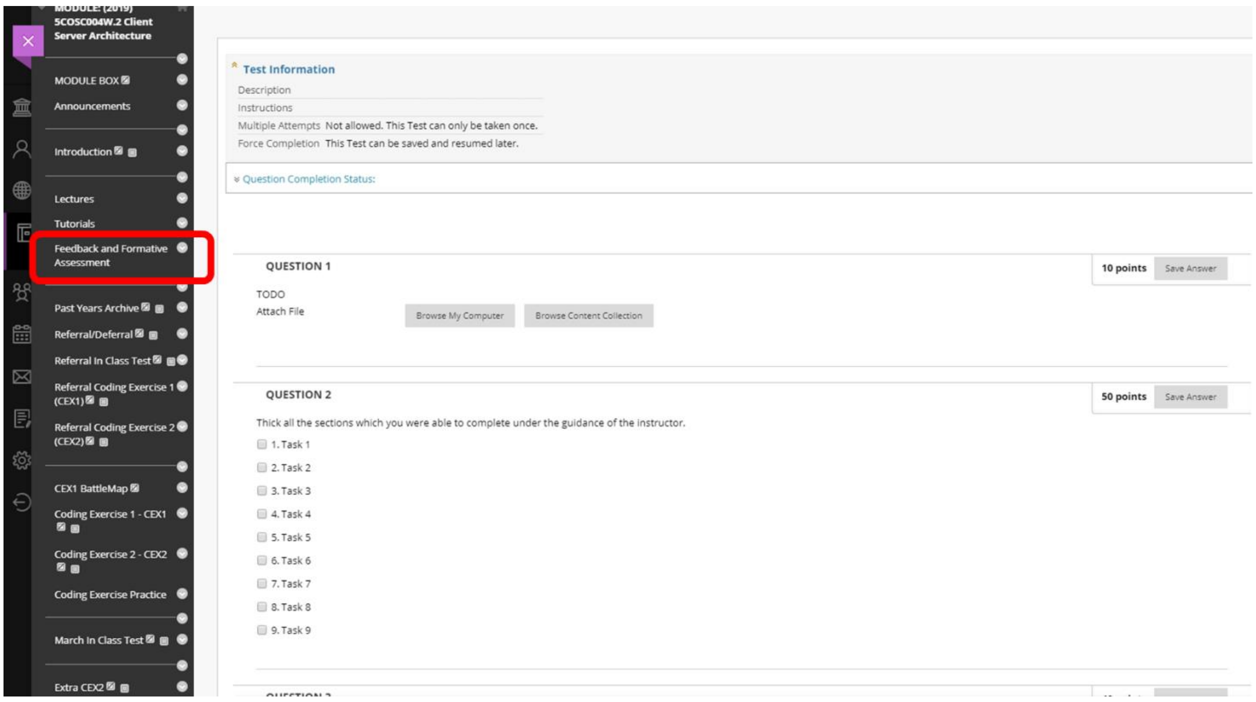
Other Directory:

Build ZIP: C:\Users\Gab\AppData\Local\Temp\SimpleTest.zip

C:\Users\Gab\AppData\Local\Temp\SimpleTest.zip already exists.

Figure 13,Export the project as zip file (2)

10) Upload the file on Blackboard and complete the self-assessment (see Figure 14)


MODULE: (2019) 5COSC004W.2 Client Server Architecture

Test Information

Description

Instructions

Multiple Attempts Not allowed. This Test can only be taken once.

Force Completion This Test can be saved and resumed later.

Question Completion Status:

QUESTION 1 — 10 points

TODO

Attach File

QUESTION 2 — 50 points

Thick all the sections which you were able to complete under the guidance of the instructor.

- 1. Task 1
- 2. Task 2
- 3. Task 3
- 4. Task 4
- 5. Task 5
- 6. Task 6
- 7. Task 7
- 8. Task 8
- 9. Task 9

Figure 14, Formative Assessment

TASKS to BE PERFORMED Independently be the student (from Task 11 to Task 15) (Formative Assessment)

11) Modify the Client so that it prints on standard output what is returned by the server.
12) Modify the testConnection method so that the client can send its id (as a string) and that is returned from the method (e.g. Connection from client ….. succeded)
13) Modify the DummyServer Class so that the server has a name (as a stgring) so that the returned string from the method testConnection is (e.g. Server….. : Connection from client ….. succeded)
14) Modify the DummyServer Class so that the server can add a time stamp to method testConnection (e.g. [Date and Time] - Server….. : Connection from client ….. succeded)
15) Upload the file on Blackboard and complete the self-assessment (see Figure 14)