Grade
/ 20

Jordan University of Science and Technology
Software Engineering Department
SE432 "Software Engineering for Web Applications"

Fall 2022, First Exam **November 5th, 2022**

Name: **ID#:** **Section:** **Serial#:**

Part 1. Short Answer Questions (14 points) [CLO1, CLO2]

1. What are the steps that are required to establish a TCP connection with the server? (**1 pt**)

   3-way handshake: First, the client asks the server to connect, the server acknowledges and asks the client to connect, then the client acknowledges the server and the connection is established.

2. I have a server that runs an HTTP server on port 80. I decided to start another HTTP server, what should be the port number of the new HTTP server? (**1 pt**)

   Any port above 1023 that is not being already reserved.

3. What does DNS stand for? (**1 pt**)

   Domain Name Service

4. Why do we need a DNS? (**1 pt**)

   To get the IP address of the host

5. Why is Transmission Control Protocol a reliable protocol? (**1 pt**)

   TCP makes sure that the packet is delivered correctly using checksum, the client needs to resubmit corrupted or missing packets. Also, the TCP guarantees that the data is received in the correct order.

6. In HTTP, what does it mean when I receive a 3xx status code? (**1 pt**)

   There is a redirection. The requested file has moved or is cached.

7. How does the HTTP server prevent the client from caching a resource? (**1 pt**)

   Using the no-store cache-control directive.

8. When using a multitier architecture, what is the difference between replication and specialization? (**1 pt**)

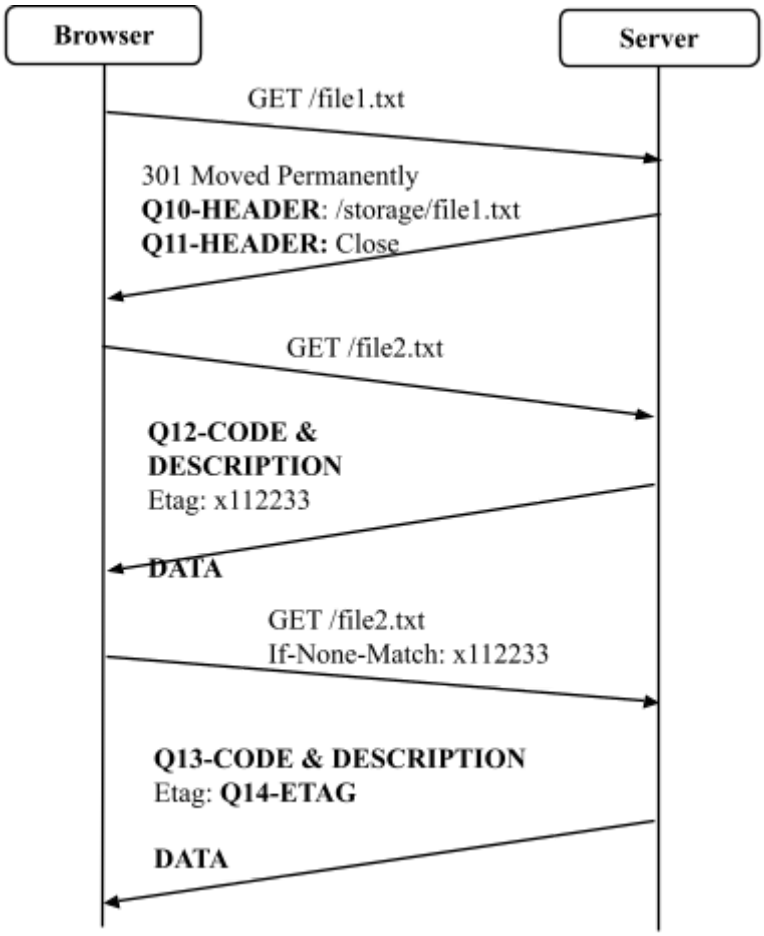
   Replication is used when we want to scale up the application by having multiple copies of the same component on different serves. Specialization is used when we want to scale the application by separating different components on different servers.

9. You have a form that contains two parameters (*course* and *semester*). The user fills *SE432* in the course field and *Fall2022* in the semester field. When submitting the form, its content is sent to the *firstexam* servlet on the server using a POST http method. The servlet runs on host www.se432.com and is listening on port 8080. What is the generated HTTP request? (**3 pts**)

   POST /firstexam HTTP/1.1
   Host: www.se432.com:8080

   Course=SE432&semester=Fall2022

Based on the following scenario, answer questions 10-14.



10. What is the value of **Q10-HEADER** (**0.5 pt**)

    Location

11. What is the value of **Q11-HEADER** (**0.5 pt**)

    Connection

12. What is the response Status Code and Description (**Q12-CODE & DESCRIPTION**) (0.5 **pt)**

    200 OK

13. What is the response Status Code and Description (**Q13-CODE & DESCRIPTION**) (**0.5 pt)**

    200 OK

14. What should be the value of **Q14-ETAG**? (1 **pt)**

    Anything other than x112233

Part 2. Servlet Programming Question (6 **points) [CLO3]**

15. You receive a *post* request that contains an unknown number of parameters. If the parameter is multivalued, then it should contain integer values. If the parameter is single-valued, then it contains a string. Write the following java code:
    a. Get the values of the parameters from the request.
    b. Add all the multivalued parameters and put their summation in a variable named *sum*.
        i. If any of the values in the list is not an integer, the program needs to ignore it. For example, if you have a multivalued parameter that has the following list:{1, 2, FOO, 4}, then you add 1, 2, and 4 and ignore the FOO value.
    c. Concatenate all the strings in the single-valued parameters in a variable called concatenatedString.

```
int sum=0; //0.25 pts
String concatenatedString =””;//0.25 pts
Enumeration params = request.getParameterNames();   //0.5 pts

while(params.hasMoreElemets()){//0.5 pts
    String param = (String)params.nextElement();//0.5 pts
    String []values = request.getParameterValues(param); //0.5 pts
    if(values.length==1){ //0.5 pts
      concatenatedString+=values[0]; //0.5 pts
    }
    else if(values.length>1}{ //0.5 pts
       for(int i=0;i<params.length;i++){//0.5 pts
           try{//0.5 pts
            sum+=Integer.parseInt(values[i]);  //0.5 pts
           }
           Catch(Exception e){ //0.5 pts
            //do nothing
           }
    }



}
```