

STEPS TO IMPLEMENT :

PART 2 :

Extract the given compressed zip folder into a regular folder. Then extract the data as in the below steps:

after

Firstly, Import the given file DSProject into the java environment. Change the path in the Server.Java file which is used to import the code from the local folder. Copy path from where it has been saved in the local device and paste as required. Run the file **Server.Java** which is the code for server. It executes and displays the output as

```
shivani@shivani-VirtualBox:~$ cd Desktop
shivani@shivani-VirtualBox:~/Desktop$ ls
Async           DSPROJECT      Synchronous
async_CS        Evaluate.class 'TableExample$1.class'
'client server' Evaluate.java   TableExample.class
client_server_1 mian_cs       TableExample.java
client_server_2 rpc
cs              rpc2
shivani@shivani-VirtualBox:~/Desktop$ cd DSPROJECT
shivani@shivani-VirtualBox:~/Desktop/DSPROJECT$ LS
LS: command not found
shivani@shivani-VirtualBox:~/Desktop/DSPROJECT$ ls
Bridge.class    client.java     part3.java    test.txt
Bridge.java     HelperThread.class server
client          part3          server.class
client.class    part3.class    server.java
shivani@shivani-VirtualBox:~/Desktop/DSPROJECT$ java server
Started Server....
```

Secondly, run the file **Client.Java** which has the code for client side with four operations namely, Upload, Delete, Rename, Download.

```
shivani@shivani-VirtualBox:~/Desktop/DSPROJECT$ java client
Enter the Type of Operation : upload or delete or rename or download
```

Next, Select an operation and type in as “upload” select the file and type in the name to upload the file into the server.

```
shivani@shivani-VirtualBox:~/Desktop/DSPROJECT$ java client
Enter the Type of Operation : upload or delete or rename or download upload
Enter Client File name
test.txt
upload is done.shivani@shivani-VirtualBox:~/Desktop/DSPROJECT$
```

Select “Download” and then give the file name to download the file to the client. The output is shown likewise:

```
PROJECTS
shivani@shivani-VirtualBox:~/Desktop/DSPROJECT$ java client
Enter the Type of Operation : upload or delete or rename or download download
Enter Server File Name to Download
test.txt
download is done
```

Run once again and now select “rename” file as input command. It asks for Enter Current File Name: Give the current file name once this step is done it shows as Enter New File Name: Give the new name that you want.

Output is like:

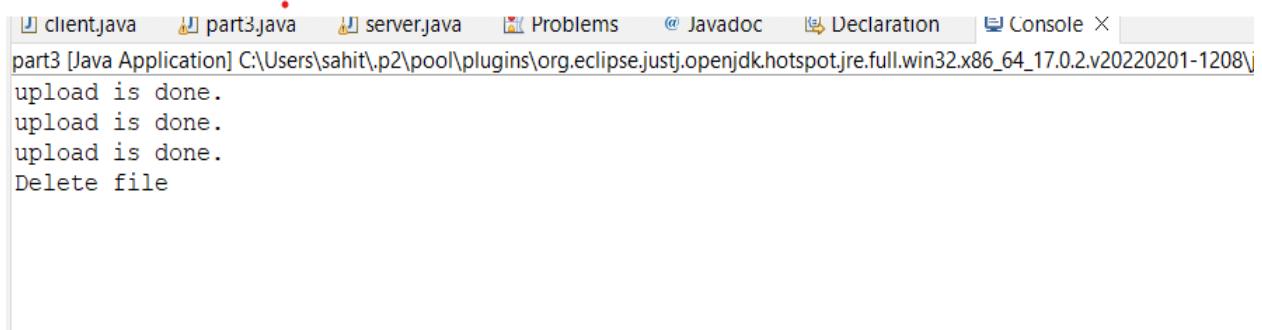
```
shivani@shivani-VirtualBox:~/Desktop/DSPROJECT$ java client
Enter the Type of Operation : upload or delete or rename or download rename
Enter Current File Name
test.txt
Enter New File Name
test1.txt
rename is done
```

Now select “delete” and the file you want to delete if the file is not found it shows an error:

```
shivani@shivani-VirtualBox:~/Desktop/DSPROJECT$ java client
Enter the Type of Operation : upload or delete or rename or download delete
Enter Server File Name
test.txt
Delete is failed
```

PART 3 :

Firstly, import the part3.Java file and execute the file. Since for the first time the code runs, the server will be considered as empty and all the files in the client will be updated in the server folder. For the next iteration(next periodic cycle). Code runs again and any changes in client will be reflected in server. When the file update, delete, or has been edited old files are stored in oldarray and new files are stored and updated in the server.



```
part3 [Java Application] C:\Users\sahit\.p2\pool\plugins\org.eclipse.jdt.openjdk.hotspot.jre.full.win32.x86_64_17.0.2.v20220201-1208\j
upload is done.
upload is done.
upload is done.
Delete file
```

PART 4 :

Synchronous:

Download the Synchronous file and unzip it. Now try to run and compile **Server.java** file by using **javac Server.java && java server** command in ubuntu terminal.

```
shivani@shivani-VirtualBox:~$ cd Desktop
shivani@shivani-VirtualBox:~/Desktop$ ls
Async           client_server_2  Evaluate.java  Synchronous
async_CS        CS              mian_cs       'TableExample$1.class'
'client server'  DSPPROJECT    rpc            TableExample.class
client_server_1 Evaluate.class  rpc2           TableExample.java
shivani@shivani-VirtualBox:~/Desktop$ cd Synchronous/
shivani@shivani-VirtualBox:~/Desktop/Synchronous$ ls
Bridge.class  Bridge.java  Client.class  Client.java  Server.class  Server.java
shivani@shivani-VirtualBox:~/Desktop/Synchronous$ javac Server.java && java Server
Started Server....
```

Now we have to open a new terminal and try to access **client.java**. Run and compile **client.java** by using **javac Client.java && java Client** command in ubuntu terminal.

```
shivani@shivani-VirtualBox:~/Desktop/Synchronous/Synchronous$ javac Client.java && java Client
Please choose any one operation : 1-> Add or 2-> Sort or 3-> Exit
```

Now the client is expected to give an input for choosing any one of the operations. For example, if a client chooses addition he goes for option 1 and enters “1” in the terminal.

```
shivani@shivani-VirtualBox:~/Desktop/Synchronous/Synchronous$ javac Client.java && java Client
Please choose any one operation : 1-> Add or 2-> Sort or 3-> Exit
1
Addition of Two Integers
Enter first number:
```

After entering option “1” it triggers the addition function in **client.java** first and performs certain operations as per request.

Now we have to enter the first number and second number to perform addition operation.

```
shivani@shivani-VirtualBox:~/Desktop/Synchronous/Synchronous$ javac Client.java && java Client
Please choose any one operation : 1-> Add or 2-> Sort or 3-> Exit
1
Addition of Two Integers
Enter first number:
3
Enter second number::
4
Result of adding the two numbers = 7
Please choose any one operation : 1-> Add or 2-> Sort or 3-> Exit
```

After generating the result , the client can choose or perform any other operation .

```
shivani@shivani-VirtualBox:~/Desktop/Synchronous/Synchronous$ javac Client.java && java Client
Please choose any one operation : 1-> Add or 2-> Sort or 3-> Exit
1
Addition of Two Integers
Enter first number:
3
Enter second number::
4
Result of adding the two numbers = 7
Please choose any one operation : 1-> Add or 2-> Sort or 3-> Exit
2
Sorting an array
Enter number of elements in the array:
4
3
5
1
2
Array elemets are:
3
5
1
2
Sorted array: 1 2 3 5
Please choose any one operation : 1-> Add or 2-> Sort or 3-> Exit
3
Program will now exit
```

Synchronous is something which schedules the task one by one so there is no overlapping of the process involved.

Asynchronous:

Let's get into Asynchronous processing.

Download the Asynchronous file and unzip it. Now try to run and compile Server.java file by using javac Server.java && java server command in ubuntu terminal.

```
...@...:~$ shivani@shivani-VirtualBox:~/Desktop/Async$ javac Server.java && java Server  
Started Server....
```

Now we have to open a new terminal and try to access client.java. Run and compile client.java by using javac Client.java && java client command in ubuntu terminal.

```
...@...:~$ shivani@shivani-VirtualBox:~/Desktop/Async$ javac Client.java && java Client  
Please choose any one operation : 1-> Add or 2-> Sort or 3-> Get Result of Operation 4-> Display  
all results 5-> Exit
```

Now the client is expected to give an input for choosing any one of the operations . For example, if a client chooses addition he goes for option 1 and enters “11” in the terminal.

```
shivani@shivani-VirtualBox:~/Desktop/Async$ javac Client.java && java Client
Please choose any one operation : 1-> Add or 2-> Sort or 3-> Get Result of Operation 4-> Display
all results 5-> Exit
1
Addition of Two Integers
Enter first number:
2
Enter second number::
3
Please choose any one operation : 1-> Add or 2-> Sort or 3-> Get Result of Operation 4-> Display
all results 5-> Exit
2
Sorting an array
Enter number of elements in the array:
3
Enter elements of the array:
3
2
1
Please choose any one operation : 1-> Add or 2-> Sort or 3-> Get Result of Operation 4-> Display
all results 5-> Exit
4
Operation Number      Result
1                      7
2                      5
3                      1 2 3
Please choose any one operation : 1-> Add or 2-> Sort or 3-> Get Result of Operation 4-> Display
all results 5-> Exit
3
```

At first the client is trying to perform addition operation and then he wants to perform some other operation immediately without knowing the result of the first one. When the second operation is performed the first task is already done by the time the client receives the result for the second task.

By this we can understand that asynchronous tasks are something which allows you to send multiple requests at a time and then can render whichever output you need of that particular task from the server.

The results or response is been stored in the table

Operation Number	Result
1	7
2	5
3	1 2 3

The first two rows in the table represents the results of add operation and the 3 rd row represents the result of sorting an array.