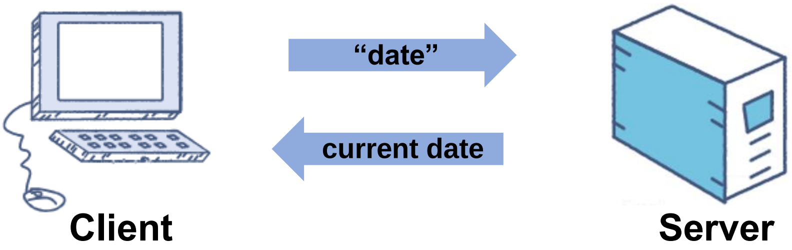
Full Stack Application Development / Software Workshop 2

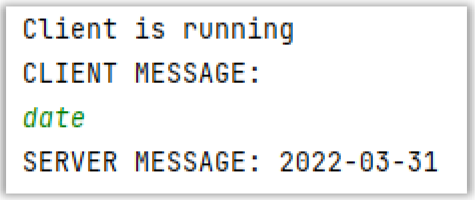
Week 11 – Java Socket Programming Lab Exercises

Exercise 1.

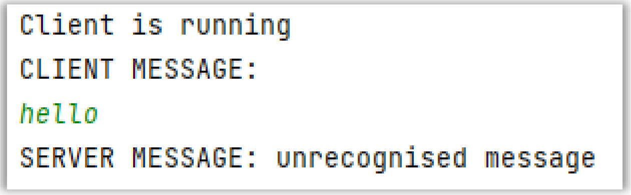
(Part a) Write a Java client-server program using TCP sockets. The client sends a message ‘date’ to a server and the server replies with the current date.



A sample program output is:



```
Client is running
CLIENT MESSAGE:
date
SERVER MESSAGE: 2022-03-31
```

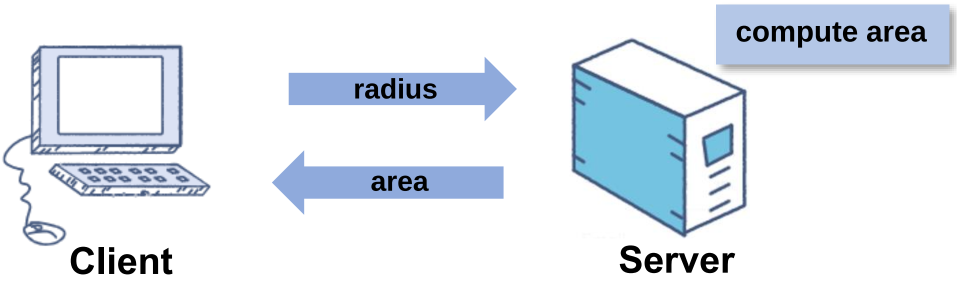


```
Client is running
CLIENT MESSAGE:
hello
SERVER MESSAGE: unrecognised message
```

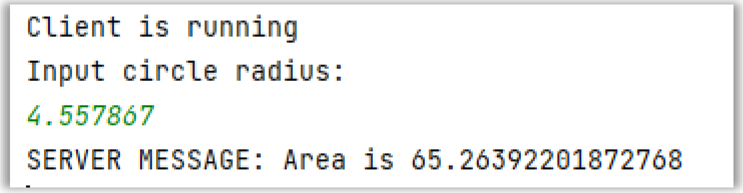
(Part b) Write the same Java program using datagram sockets.

Exercise 2.

(Part a) Write a Java client-server program using TCP sockets. The client sends the radius of a circle to the server and the server computes the area of the circle and returns it to the client.



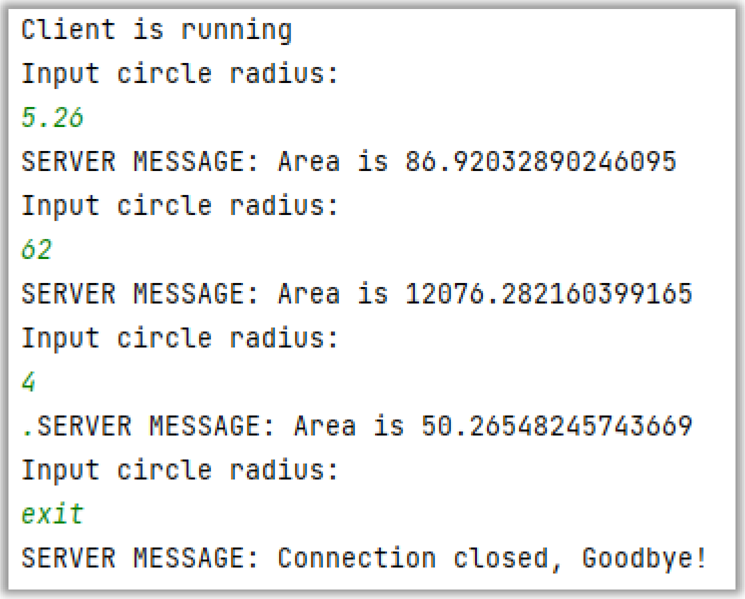
A sample program output is:



```
Client is running
Input circle radius:
4.557867
SERVER MESSAGE: Area is 65.26392201872768
```

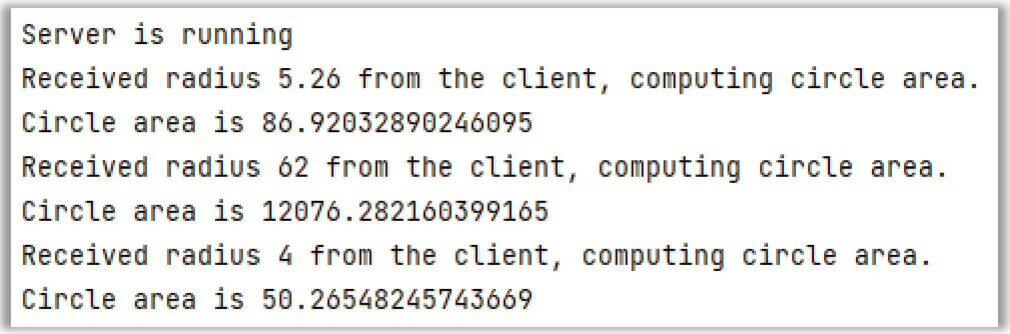
(Part b) Modify the above program such that it provides a continuous communication between the client and the server. The connection closes when the client sends the ‘exit’ message.

A sample program output is:



```
Client is running
Input circle radius:
5.26
SERVER MESSAGE: Area is 86.92032890246095
Input circle radius:
62
SERVER MESSAGE: Area is 12076.282160399165
Input circle radius:
4
.SERVER MESSAGE: Area is 50.26548245743669
Input circle radius:
exit
SERVER MESSAGE: Connection closed, Goodbye!
```

Client output

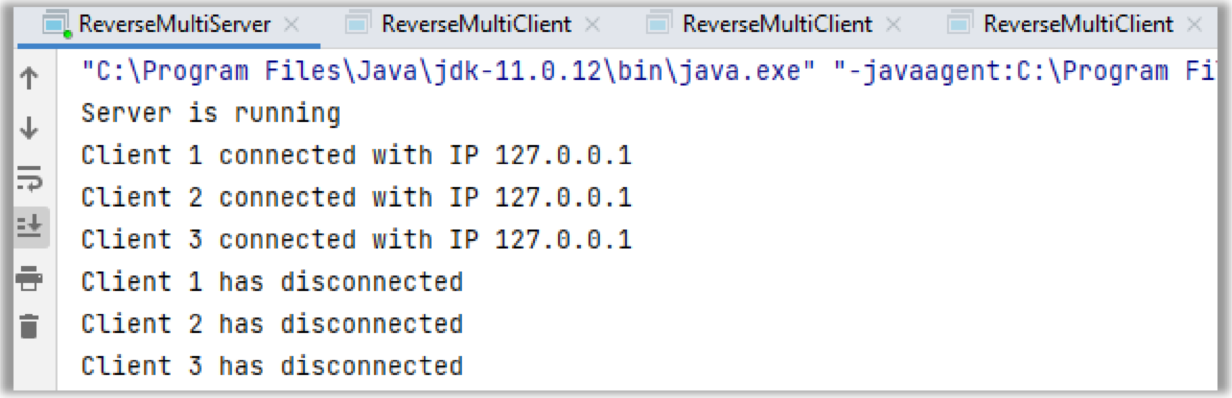


```
Server is running
Received radius 5.26 from the client, computing circle area.
Circle area is 86.92032890246095
Received radius 62 from the client, computing circle area.
Circle area is 12076.282160399165
Received radius 4 from the client, computing circle area.
Circle area is 50.26548245743669
```

Server output

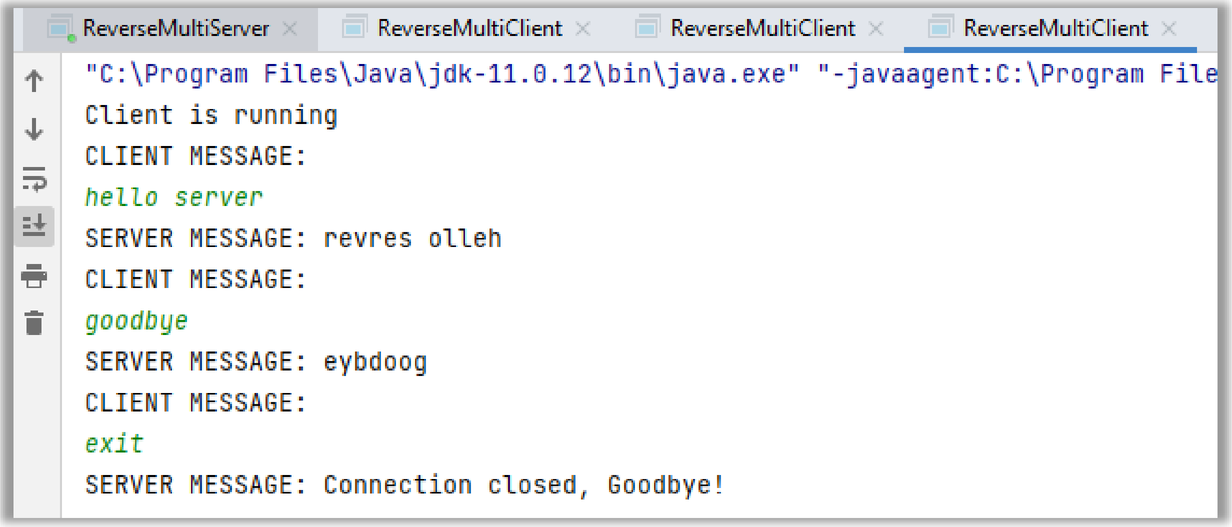
Exercise 3. Write a TCP multithreaded client-server application in Java. The client continuously sends messages to the server, until the user types “exit”. The server reverses the received message and sends it back to the client.

A sample output program with three clients is:



```
"C:\Program Files\Java\jdk-11.0.12\bin\java.exe" "-javaagent:C:\Program Fi
Server is running
Client 1 connected with IP 127.0.0.1
Client 2 connected with IP 127.0.0.1
Client 3 connected with IP 127.0.0.1
Client 1 has disconnected
Client 2 has disconnected
Client 3 has disconnected
```

Server output



```
"C:\Program Files\Java\jdk-11.0.12\bin\java.exe" "-javaagent:C:\Program File
Client is running
CLIENT MESSAGE:
hello server
SERVER MESSAGE: revres olleh
CLIENT MESSAGE:
goodbye
SERVER MESSAGE: eybdoog
CLIENT MESSAGE:
exit
SERVER MESSAGE: Connection closed, Goodbye!
```

Client output