

## The Assignment 2 Specification and Marking Criteria

### Peer-to-Peer Content Distribution and Distributed Query

In distributed computing, a peer is both a client and a server at the same time, being able to request services from other peers or providing services to other peers. For the general knowledge about peer, you can read the following academic article.

Rodrigues, R., Druschel, P., 2010. *Peer-to-Peer Systems*, *Communications of the ACM*. 53(10), 72-82. (available at <http://zoo.cs.yale.edu/classes/cs426/2012/bib/rodrigues10peer-to-peer.pdf> on 23rd Dec 2017).

In this assignment, you are to develop programming skills of peer via a particular Peer-to-Peer (P2P) application – *content distribution and distributed query*. The application allows peers to form an overlay and share information from the entire peer overlay. In this assignment, you will need to upgrade a client and a server into a peer. The peers are to form an overlay by IP multicast and the information stored in a peer can be shared by any other peers. When a peer queries a question, it will multicast a request including the question to the peer overlay and the peer that has the answer to the question to respond to the requester peer. To query a question, a UDP datagram via IP multicast is used. Once the answer is found, replying the answer is to use another separate UDP communication between the requester and the replier. To complete this assignment, you need to complete the following 2 parts.

#### Part 1: Peer overlay design and implementation

In this part, you are to develop the simplified system of content distribution and distributed query. The assumptions on the system are specified as follows.

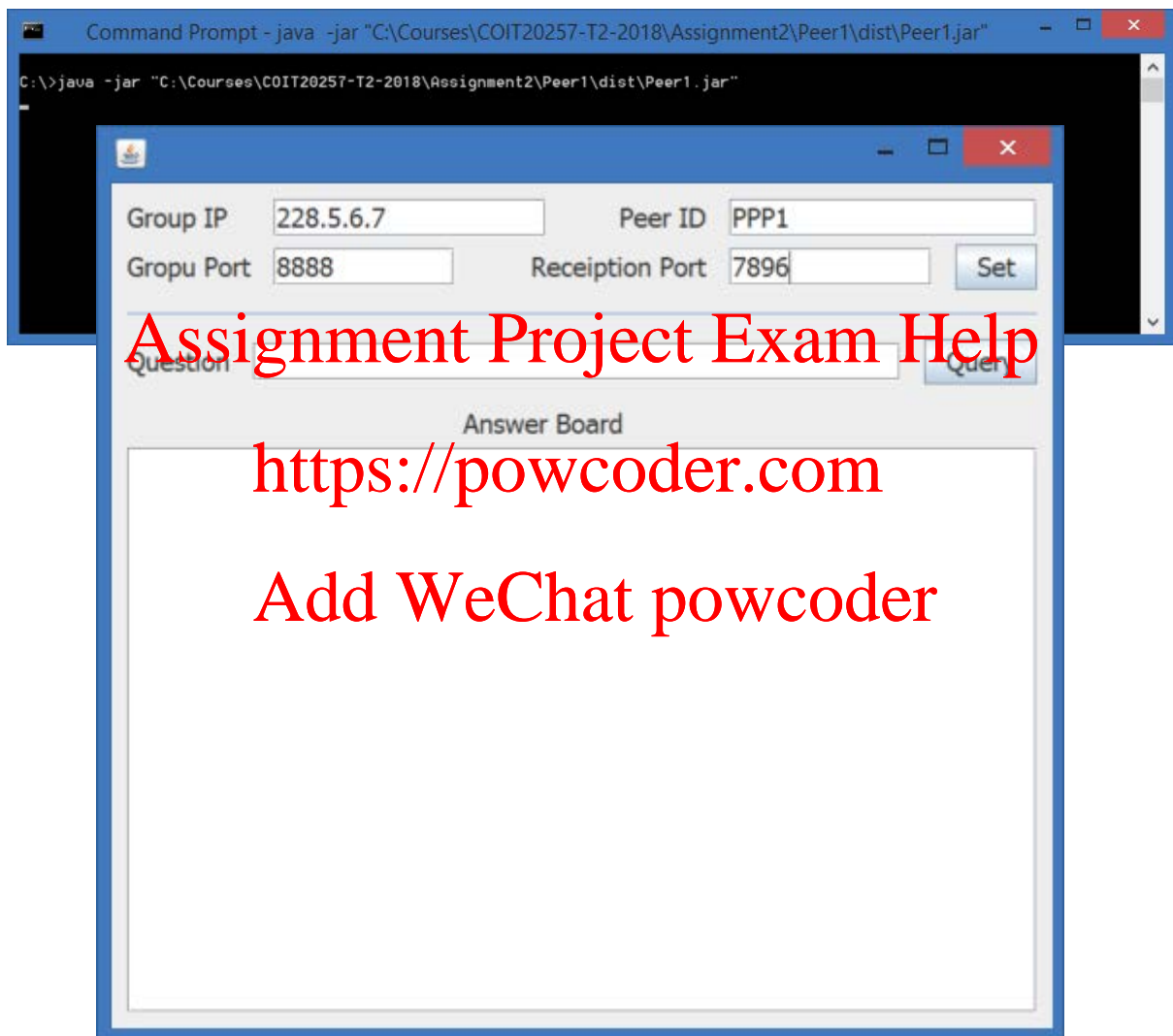
1. Each peer has a unique ID.
2. Peers form a networking overlay by IP multicast (reference to Chapter 4.4 of the textbook, Week 5 lecture slides, Week 6 lab projects, and search for other academic articles about IP multicast if necessary).
3. Each peer has a number of answers to some questions that can be shared by other peers.
4. Once a peer has a question, it needs to compose a request, including the question and communication information such as reply IP address and port number.
5. Each peer can multicast a query request to all other peers in the overlay. After that, the peer is waiting on the port for answers.
6. To simplify the problem, we assume that the answer to a queried question can always be found from at least one peer.
7. Once the answer to the question is found, the replier peer (the one who has the answer) will use a UDP datagram to send the answer to the requester peer (the one who asks the question).

8. A peer needs to ignore the requests from itself because by default IP multicast sends a message to every peer on the overlay.

The following screenshots show an implementation of the system of content distribution and distributed query.

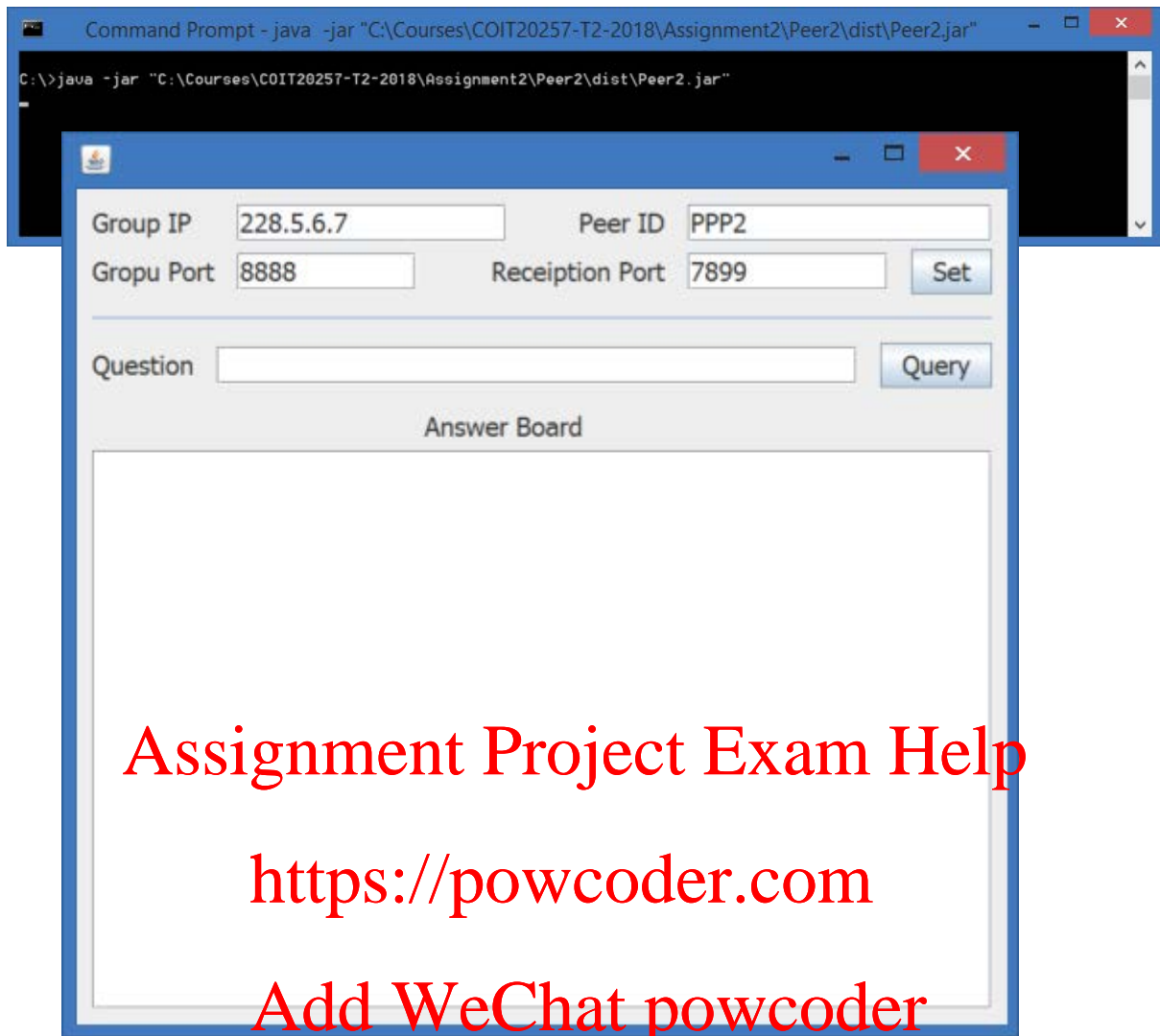
1. The 3 peers (with ID of *PPP1*, *PPP2* and *PPP3*) join the same IP multicast group. Note: 228.5.6.7 and 8888 are the IP multicast address and port number that were used in this example.

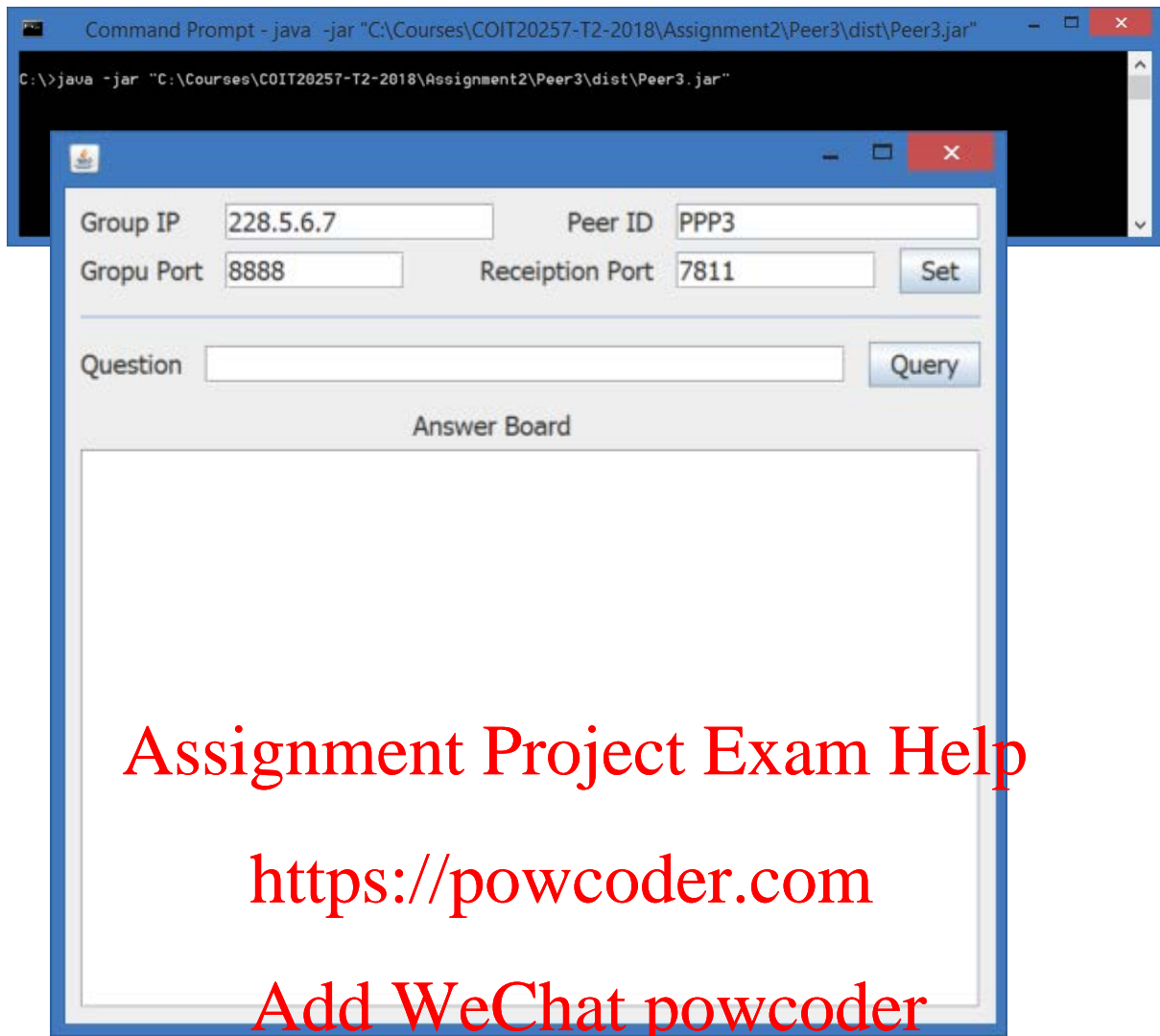
PPP1 joins (using port 7896 as the UDP port to receive answers)



In the above example, clicking the Set button will bring *PPP1* into full function by joining the multicast group and initialising the service threads for querying questions and receiving answers.

*PPP2* and *PPP3* start and join the peer overlay in the same way as shown by the following screenshots.





2. The following screenshots show the single-answer examples:

*PPP1* queries Question 2, only *PPP2* has the answer and replies. Note: entering a question on *PPP1*'s Question text field and clicking the Query button will multicast the question to the peer overlay and query to all the peers.

*PPP2* queries Question 3, only *PPP3* has the answer and replies.

*PPP3* queries Question 1, only *PPP1* has the answer and replies.

Content Distribution and Distributed Query - PPP1

Group IP: 228.5.6.7 Peer ID: PPP1

Group Port: 8888 Reception Port: 7896 Set

Question: Question 2 Query

Answer Board

PPP2: Answer 2 (for Question 2)

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Content Distribution and Distributed Query - PPP2

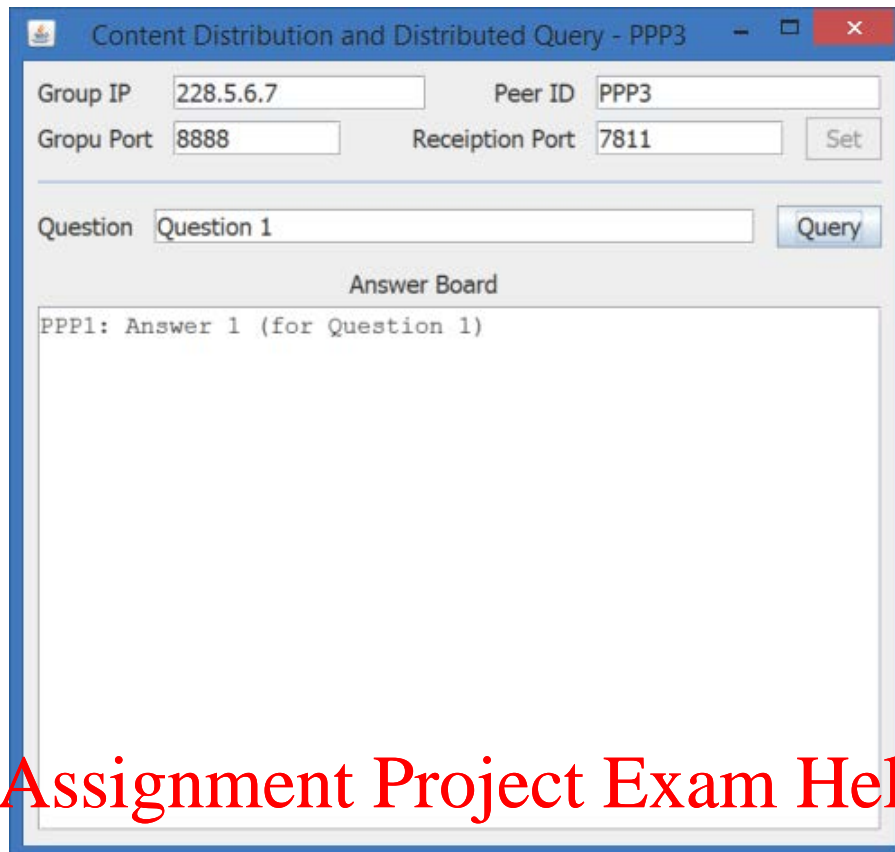
Group IP: 228.5.6.7 Peer ID: PPP2

Group Port: 8888 Reception Port: 7899 Set

Question: Question 3 Query

Answer Board

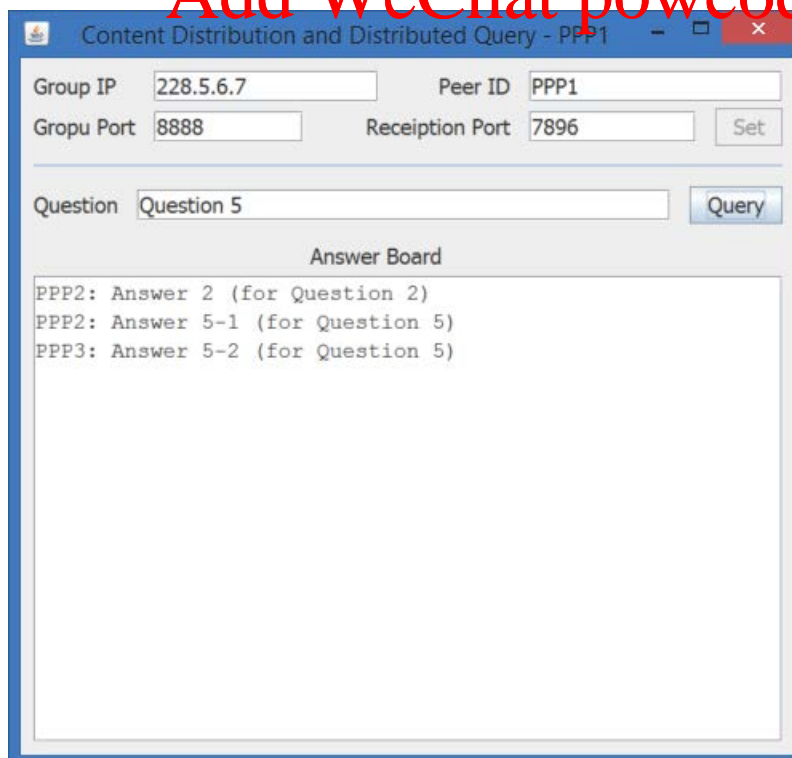
PPP3: Answer 3 (for Question 3)



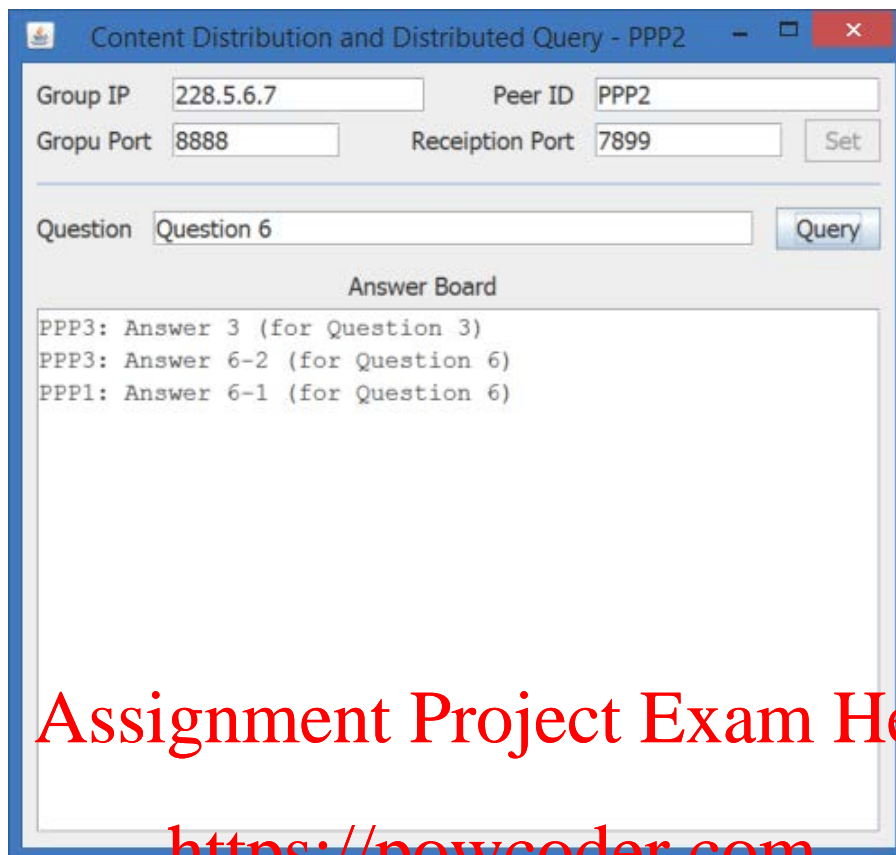
## Assignment Project Exam Help

3. The following screenshots show the multiple answer examples. Note the multiple-answer function is required by this assignment.

PPP1 queries Question 5, PPP2 and PPP3 have answers and reply.



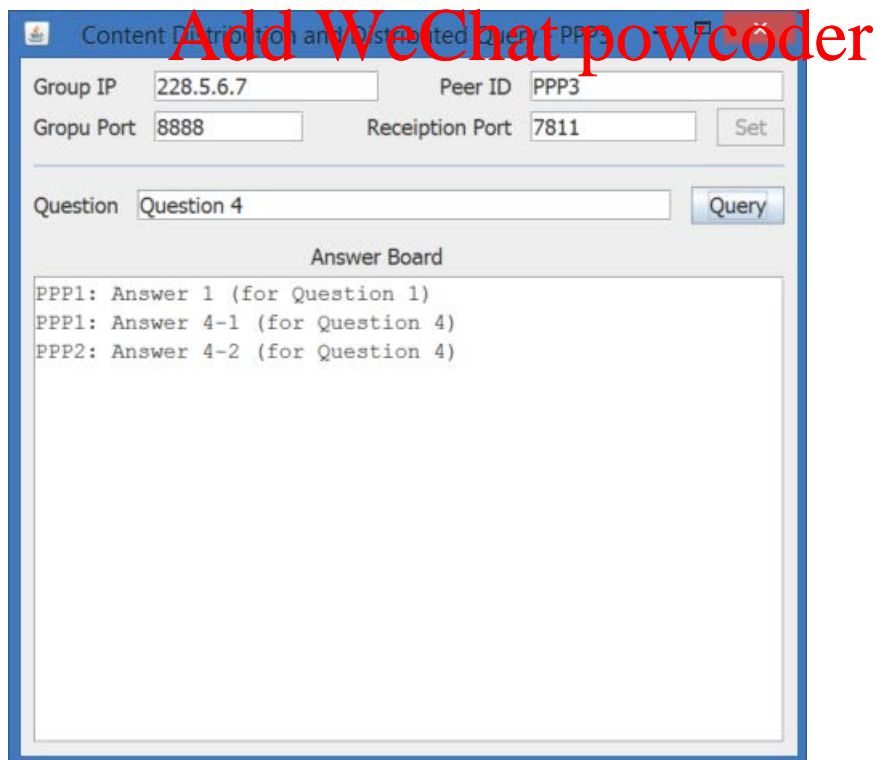
PPP2 queries Question 6, PPP1 and PPP3 have answers and reply.



Assignment Project Exam Help

<https://powcoder.com>

PPP3 queries Question 4, PPP1 and PPP2 have answers and reply



Add WeChat powcoder

The above example shows the equity between peers, that is, **every peer is both a client and a server to share contents and distributed processing**. Your implementation must satisfy all the above requirements in order to get the full mark.

## Part 2: Documentation

On completion of the implementation of the framework, prepare a document to include:

1. An end user' instruction about how to compile, run and test your system.
2. The key issues about upgrading a client and a server to a peer and overlay forming.

## Submission

You need to provide the following files in your submission.

1. At least 3 peers in 3 individual folders, of which **each folder** contains:
  - Files of Java source code of the implementation. The in-line comments on the data structure and program structure in the programs are required. These source code files must be able to be compiled by the standard JDK (Java Development Kit) or NetBeans from Oracle (<http://www.oracle.com/technetwork/java/index.html>)
  - The compiled Java class files of the source code. These Java classes must be runnable on the standard JRE (Java Runtime Environment) or NetBeans from Oracle (<http://www.oracle.com/technetwork/java/index.html>)
2. A Microsoft Word document to address the issues as specified in Part 2 above.

All the required files must be compressed into a zip file for submission. You must submit your assignment via the online submission system from the unit web site. **Any hardcopy or email submission will not be accepted. After the marked assignments are returned, any late submissions will not be accepted.**

## The Marking Criteria

Marking Criteria	Available Marks
<b>Part 1: Peer overlay design and implementation</b>	<b>29</b>
1. Whether the peer is able to take the role of being both a client and a server	2
2. Whether the general structure of peer is sound	2
3. Whether the overlay in terms of 2 communication channels, one for multicast and the other for answer reception, is structured correctly	2



4. Whether a peer is able to create and join the IP multicast group	2
5. Whether a peer is able to multicast a query request by using a UDP datagram	2
6. Whether the request encapsulates a peer's ID and UDP reception port and a question	3
7. Whether the request from a peer itself is ignored	2
8. Whether multiple-answer function is correctly implemented	3
9. Whether a service thread is used when necessary	2
10. Whether the application is compilable and runnable	2
11. Whether the 3 test peers are provided	3
12. Whether the source code is readable	2
13. Whether necessary in-line comments are provided	2
<b>Part 2: Documentation</b>	<b>6</b>
1. Whether the instruction covers compiling and starting of the application	2
2. Whether the instruction covers the tests of both the single-answer and multi-answer cases	2
3. Whether the key issues about upgrading a client and a server to a peer and overlay forming are addressed	2
<b>Sub Total for Assignment 2</b>	<b>35</b>
<b>Late Penalty</b>	-1.75 (5% each calendar day, either full or partial)
<b>Plagiarism Related Penalty</b>	
<b>Total for Assignment 2</b>	