NAME: SHASHWAT SHAH
SAP ID: 60004220126
DIV/BATCH: C22

DISTRIBUTED COMPUTING (DC)
EXPERIMENT 09

AIM: To demonstrate mutual exclusion algorithm using JAVA.

CODE:
MutualServer.java

```
import java.io.*;
import java.net.*;

public class MutualServer implements Runnable {
   Socket socket = null;
   static ServerSocket ss;

   // Constructor to initialize socket
   MutualServer(Socket newSocket) {
      this.socket = newSocket;
   }

   public static void main(String[] args) throws IOException {
      ss = new ServerSocket(7000);
      System.out.println("Server Started...");

      while (true) {
         Socket s = ss.accept();
         System.out.println("New client connected...");
         MutualServer es = new MutualServer(s);
         Thread t = new Thread(es);
         t.start();
      }
   }

   // Runnable method to read data from client
   public void run() {
      try {
         BufferedReader br = new BufferedReader(new InputStreamReader(socket.getInputStream()));
         while (true) {
            String str = br.readLine();
            if (str != null) {
               System.out.println("Client: " + str);
            }
         }
      } catch (Exception e) {
         System.out.println("Exception: " + e);
      }
   }
}
```

ClientOne.Java

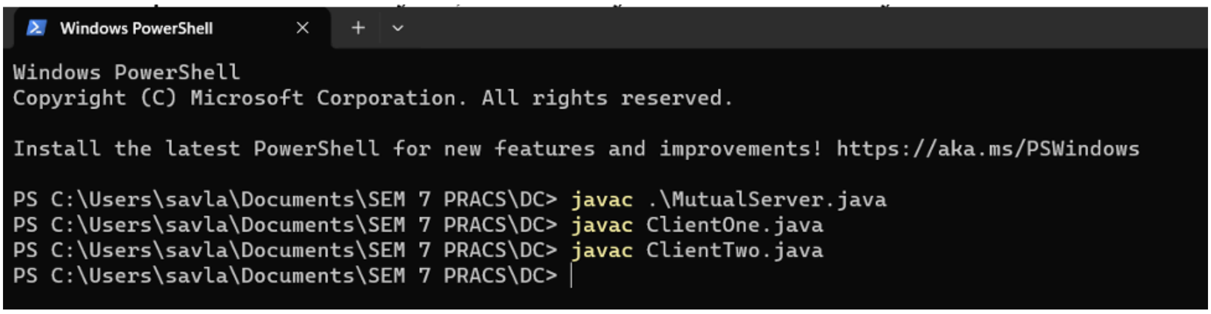
```
import java.io.*;
import java.net.*;

public class ClientOne {
    public static void main(String[] args) throws IOException {
        // Connect to MutualServer on port 7000
        Socket s = new Socket("localhost", 7000);
        PrintStream out = new PrintStream(s.getOutputStream());

        // Create a server socket to receive the token from ClientTwo
        ServerSocket ss = new ServerSocket(7001);
        Socket s1 = ss.accept();
        BufferedReader in1 = new BufferedReader(new InputStreamReader(s1.getInputStream()));
        PrintStream out1 = new PrintStream(s1.getOutputStream());
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

        String str = "Token";
        while (true) {
            if (str.equalsIgnoreCase("Token")) {
                System.out.println("Do you want to send some data?");
                System.out.println("Enter Yes or No");
                str = br.readLine();

                if (str.equalsIgnoreCase("Yes")) {
                    System.out.println("Enter the data");
                    str = br.readLine();
                    out.println(str); // Send data to MutualServer
                }

                // Pass the token to ClientTwo
                out1.println("Token");
            }

            System.out.println("Waiting for Token");
            str = in1.readLine();
        }
    }
}
```

ClientTwo.Java

```
import java.io.*;
import java.net.*;

public class ClientTwo {
    public static void main(String[] args) throws IOException {
        // Connect to MutualServer on port 7000
        Socket s = new Socket("localhost", 7000);
        PrintStream out = new PrintStream(s.getOutputStream());
        // Connect to ClientOne's server socket on port 7001 to receive and send the token
        Socket s2 = new Socket("localhost", 7001);
        BufferedReader in2 = new BufferedReader(new InputStreamReader(s2.getInputStream()));
        PrintStream out2 = new PrintStream(s2.getOutputStream());
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

        String str = "Token";
        while (true) {
          System.out.println("Waiting for Token");
          str = in2.readLine(); // Wait for the token from ClientOne

          if (str.equalsIgnoreCase("Token")) {
            System.out.println("Do you want to send some data?");
            System.out.println("Enter Yes or No");
            str = br.readLine();

            if (str.equalsIgnoreCase("Yes")) {
              System.out.println("Enter the data");
              str = br.readLine();
              out.println(str); // Send data to MutualServer
            }

            // Pass the token back to ClientOne
            out2.println("Token");
          }
        }
    }
}
```

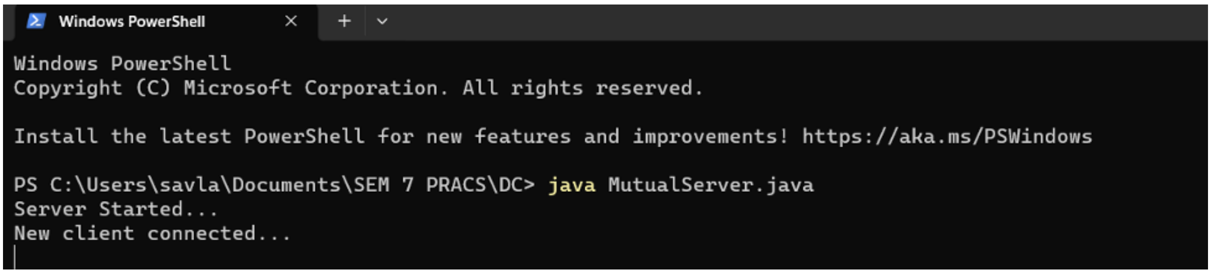
OUTPUT:

- **Compile MutualServer.java, ClientOne.java and ClientTwo.java**



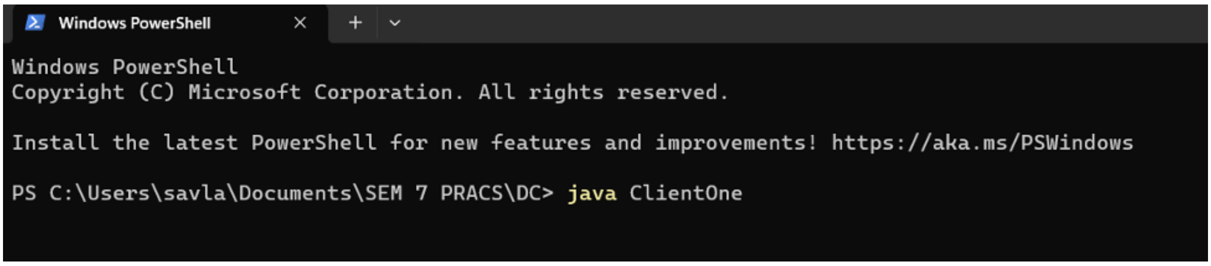
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\savla\Documents\SEM 7 PRACS\DC> javac .\MutualServer.java
PS C:\Users\savla\Documents\SEM 7 PRACS\DC> javac ClientOne.java
PS C:\Users\savla\Documents\SEM 7 PRACS\DC> javac ClientTwo.java
PS C:\Users\savla\Documents\SEM 7 PRACS\DC>
```

- **Execute MutualServer.java**



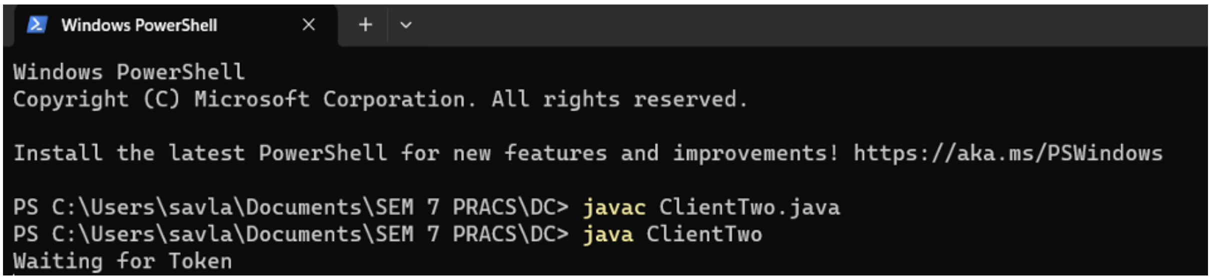
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\savla\Documents\SEM 7 PRACS\DC> java MutualServer.java
Server Started...
New client connected...
```

- **Open a new command prompt and execute ClientOne.java. Keep it running till ClientTwo starts**

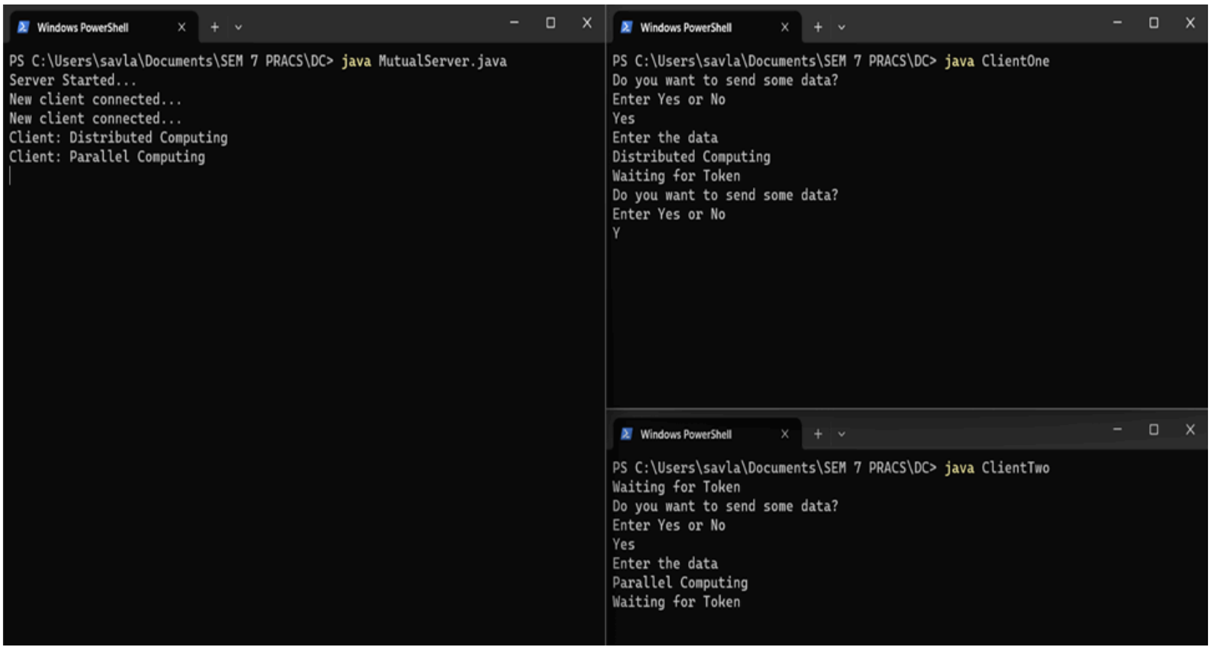


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\savla\Documents\SEM 7 PRACS\DC> java ClientOne
```

- **Open another command prompt and execute ClientTwo.java. The output allows both the clients to use tokens and share their messages with each other using Token Ring concept. To send the message, the client has to accept the token by typing Yes followed by the message alternately and has to type No to release the token**



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\savla\Documents\SEM 7 PRACS\DC> javac ClientTwo.java
PS C:\Users\savla\Documents\SEM 7 PRACS\DC> java ClientTwo
Waiting for Token
```



```
PS C:\Users\savla\Documents\SEM 7 PRACS\DC> java MutualServer.java
Server Started...
New client connected...
New client connected...
Client: Distributed Computing
Client: Parallel Computing
```

```
PS C:\Users\savla\Documents\SEM 7 PRACS\DC> java ClientOne
Do you want to send some data?
Enter Yes or No
Yes
Enter the data
Distributed Computing
Waiting for Token
Do you want to send some data?
Enter Yes or No
Y
```

```
PS C:\Users\savla\Documents\SEM 7 PRACS\DC> java ClientTwo
Waiting for Token
Do you want to send some data?
Enter Yes or No
Yes
Enter the data
Parallel Computing
Waiting for Token
```