Program 7: Using TCP/IP sockets, write a client – server program to make the client send the file name and to make the server send back the contents of the requested file if present.

TCPC.java

```
import java.net.*;  //Networking classes like Socket.
import java.io.*;  //Input/Output streams for reading and writing data.

public class TCPC {
   public static void main(String[] args) throws Exception {
      // Step 1: Connect to the server at localhost on port 4000
      Socket sock = new Socket("127.0.0.1", 4000);  //each socket has inputstream and outstream
                                                                  (receive)          (send)
      // Step 2: Read filename input from user
      System.out.println("Enter the file name:");.
      BufferedReader keyRead = new BufferedReader(new InputStreamReader(System.in));
      String fname = keyRead.readLine(); //stores entered filename into fname

      // Step 3: Send the filename(fname) to the server
      OutputStream ostream = sock.getOutputStream();
      PrintWriter pwrite = new PrintWriter(ostream, true);
      pwrite.println(fname);                  //BufferedReader  lets you read text easily (line by line).
                                              //InputStreamReader: converts bytes to characters

     // Step 4: Receive and display file contents from the server
      InputStream istream = sock.getInputStream();
      BufferedReader socketRead = new BufferedReader(new InputStreamReader(istream));

      String str;
      while ((str = socketRead.readLine()) != null) {
         System.out.println(str);
      }

      // Step 5: Close resources
      pwrite.close();
      socketRead.close();
      keyRead.close();
      sock.close();
   }
}
```

TCPS.java

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

public class TCPS {
   public static void main(String args[]) throws Exception {
      // Step 1: Start server on port 4000
      ServerSocket sersock = new ServerSocket(4000);
      System.out.println("Server ready for connection!!");

      // Step 2: Accept client connection
      Socket sock = sersock.accept();
      System.out.println("Connection is successful and waiting for file name...");

      // Step 3: Read the filename sent by the client
      InputStream istream = sock.getInputStream();
      BufferedReader fileRead = new BufferedReader(new InputStreamReader(istream));
      String fname = fileRead.readLine();

     // Step 4: Open and read the file
      BufferedReader contentRead = new BufferedReader(new FileReader(fname));

      // Step 5: Send file content to client
      OutputStream ostream = sock.getOutputStream();
      PrintWriter pwrite = new PrintWriter(ostream, true);

      String str;
      while ((str = contentRead.readLine()) != null) {
         pwrite.println(str);
      }

      // Step 6: Close all connections
      pwrite.close();
      contentRead.close();
      fileRead.close();
      sock.close();
      sersock.close();
   }
}
```

sample.java

```
hi
hello
```