```
// ***** tcpserver.cpp *****

//*********************************************************************
//
// Computer Science 4/5313 Computer Networks
//
// Spring  2016
//
// Instructor:  Hung-Chi Su
//
// Assignment # n
//
// Programmer:  your name
//
// Due Date:    day-of-week, month day, year
//
// Description: This is a TCP socket program that illustrates how a server
//              take one message from a client
//
//  Input:      from client via TCP connection, and echo back to client
//
//  Output:     message from client is written to the screen.
//
//  Compile:    g++ -o tcpserver1 tcpserver.cpp
//
//  Command:    ./tcpserver1 [<port>]
//
//  Note:       Change the last 4 digits of  MY_PORT_ID to
//              the last 4 digits of your student id
//*********************************************************************
                    // header files

#include <stdio.h>      //for printf(), ...
#include <stdlib.h>
#include <string.h>

//#include <sys/types.h>        //for data types
#include <sys/socket.h> //for socket(), connect(), ...
#include <unistd.h>     //for close()
#include <netinet/in.h> //for internet address family
#include <arpa/inet.h>  //for sockaddr_in, inet_addr()
#include <errno.h>      //for system error numbers
#include <iostream>     //for cin, cout...

// !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
// change the last 4 digits to your last 4 digits of student ID
// !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
#define MY_PORT_ID   11075      // or 21234, 31234, 41234, 51234
```

```
#define MESSAGESIZE  50

using std::cout;
using std::cerr;

main( int argc, char *argv[])
{
  int sockfd, newsockfd,          // file descriptor of sockets
     nread,                       // the # of bytes read
     addrlen, client_addrlen;    // address length
  int my_port_id = MY_PORT_ID;
  struct sockaddr_in my_addr, client_addr;  // addresses for socket
  char msg[MESSAGESIZE];

  if (argc == 2)
  {
     my_port_id = atoi(argv[1]);
  }

  // ------------------------------------------------------
  // Initialization:
  // ------------------------------------------------------
  //cout << "Server: creating socket\n";
  //if ( (sockfd = socket(PF_INET, SOCK_STREAM, 0)) < 0)
  if ( (sockfd = socket(AF_INET, SOCK_STREAM, 0)) < 0)
  {
    cerr << "Server: socket error: " << errno << "\n";
    exit(1);
  }

  // cout << "Server: constructing my local address\n";
  memset( &my_addr, 0, sizeof(my_addr));    // Zero out structure
  my_addr.sin_family = AF_INET;        // Internet address family
  my_addr.sin_addr.s_addr = htonl(INADDR_ANY);    // Any incoming interface
  my_addr.sin_port = htons(my_port_id);    // my port


  // ------------------------------------------------------
  // binding:
  // ------------------------------------------------------
  if ( (bind(sockfd, (struct sockaddr *) &my_addr, sizeof(my_addr)) < 0) )
  {
    cerr << "Server: bind fail: " << errno << "\n";
    exit(2);
  }
  // ------------------------------------------------------
  // Tell the OS for this socket to take message
  // ------------------------------------------------------
  //listen(sockfd, 5);  // for 5 pending request only
```

```
  listen(sockfd, SOMAXCONN);  //  decide by OS
  cout << "\n\nWaiting for client's message....\n\n";

  // Loop
  client_addrlen = sizeof(client_addr);

  while(1)
  {
    // ------------------------------------------------------
    // Wait for client's connection
    // ------------------------------------------------------
    newsockfd = accept(sockfd, (struct sockaddr *) &client_addr, (socklen_t *)&client_addrlen);
    if (newsockfd < 0)
    {
      cerr << "Server: accept error: " << errno << "\n";
      exit(2);
    }

    cout << "Connection from relay(" << inet_ntoa(client_addr.sin_addr);  //client IP
    cout << ":" << ntohs(client_addr.sin_port) << ")\n"; // client port

    nread = read(newsockfd,msg, MESSAGESIZE);
    // cout << "Server: retrun code from read is " << nread << "\n";
    if (nread >0)
    {

      cout << "Relay's message is: ";
      cout.write(msg, nread);                  // Message from client
      // cout << msg ;  // incorrect way to do
      cout << "\n";
    }

    // echo back
    nread = write(newsockfd,msg,strlen(msg));
    if (nread < 0)
    {
      cerr << "server's send failed...\n";
      exit(3);
    }

    close(newsockfd);
  }
  // ------------------------------------------------------
  // Termination
  // ------------------------------------------------------
  /* close socket */
  close(sockfd);
}
```