Sheet 1

Simple client program in C

```
/*
 * This is a very simple client program designed to interact with an
 * equivalent server. It creates a message, sends it to a server, and
 * awaits a reply.
 */

#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <stdio.h>
#include <stdlib.h>

#define BUFFSIZE 150

int main()
{ struct sockaddr_in serv;
  char buf[BUFFSIZE];
  int  sockfd, n;

  // Fill in the buffer with something sensible
  //
  strcpy(buf, "Hello there");

  // Now create a datagram (i.e. UDP) socket. This returns
  // a descriptor used in subsequent calls
  //
  if ((sockfd = socket(PF_INET, SOCK_DGRAM, 0)) < 0)
  { perror("socket error"); return -1; }

  // We're going to use the socket to send first. Create a structure
```

```
   // to hold the server's address.
   //
   bzero ( (char *)&serv, sizeof(serv) );
   serv.sin_family        = AF_INET;
   serv.sin_addr.s_addr   = inet_addr("128.16.6.210");
   serv.sin_port          = htons(13);

   // Now send the datagram using the structure defined above
   //
   if (sendto(sockfd, buf, BUFFSIZE, 0,
          (struct sockaddr *)&serv, sizeof(serv)) != BUFFSIZE)
   { perror("sendto error"); return -1; }

   // And wait for a reply from the server
   //
   if ((n = recvfrom(sockfd, buf, BUFFSIZE, 0,
             (struct sockaddr *)NULL, (int *)NULL)) < 2)
   { perror("recvfrom error"); return -1; }

   // If we're going to print this as a string, need to put in the
   // terminating 0
   //
   buf[n-2] = 0;
   printf("%s\n", buf);

   exit(0);
 }
```