# TASK 1:

**Server.cpp:**

﻿﻿#include<iostream>

#include <sys/types.h>

#include <sys/socket.h>

#include <netinet/in.h>

#include <arpa/inet.h>

#include <stdlib.h>

#include <unistd.h>

using namespace std;

int main()

{

struct in\_addr addr;

char msg [100];

int conn\_sock, comm\_sock,n;

struct sockaddr\_in server\_addr, client\_addr;

conn\_sock= socket (AF\_INET, SOCK\_STREAM, 0);

server\_addr.sin\_family=AF\_INET;

server\_addr.sin\_port=1234;

server\_addr.sin\_addr.s\_addr=inet\_addr ("127.0.0.1");

bind (conn\_sock, (struct sockaddr \*) &server\_addr, sizeof (server\_addr));

listen (conn\_sock, 10);

comm\_sock = accept (conn\_sock, (struct sockaddr \*) &client\_addr, (socklen\_t \*)&client\_addr);

cout<<"\n\nConnection established with client";

n=read (comm\_sock, msg,100);

cout<<"\n\nMessage received: "<< msg << endl;

close (comm\_sock);

close (conn\_sock);

return 0;

}

**Client.cpp:**

#include <iostream>

#include <sys/socket.h>

#include <netinet/in.h>

#include <arpa/inet.h>

#include <cstring>

#include <unistd.h>

using namespace std;

int main()

{

int client\_sock;

struct sockaddr\_in server\_addr;

char msg[100];

client\_sock = socket(AF\_INET, SOCK\_STREAM, 0);

server\_addr.sin\_family = AF\_INET;

server\_addr.sin\_port = 1234;

server\_addr.sin\_addr.s\_addr = inet\_addr("127.0.0.1");

connect(client\_sock, (struct sockaddr\*)&server\_addr, sizeof(server\_addr));

cout << "Connected to the server." << endl;

cout << "Enter a message to send to the server: ";

cin.getline(msg, sizeof(msg));

send(client\_sock, msg, strlen(msg), 0);

cout << "Message sent to the server." << endl;

close(client\_sock);

return 0;

}

# OUTPUT:

