ABES Institute of Technology, Ghaziabad

COLLEGE CODE – 290

Lab File

NAME	SANDEEP KUMAR SHUKLA
BRANCH	CSE
UNIVERSITY ROLL NO.	1729010140
SESSION	2020-21
NAME OF LAB	Distributed System Lab (RCS 751)

Aim :- *Write a program to simulate the functioning of Lamport's Logical clock in 'C'.*

Code :-

```
#include <stdio.h>
int max1(int a, int b)
{
   if (a > b)
      return a;
   else
      return b;
}
int main()
{
   int i, j, k, p1[20], p2[20], e1, e2, dep[20][20];
   printf("Enter the events : ");
   scanf("%d %d", &e1, &e2);
   for (i = 0; i < e1; i++)
      p1[i] = i + 1;
   for (i = 0; i < e2; i++)
      p2[i] = i + 1;
   printf("Enter the dependency matrix:\n");
   printf("\t Enter 1 if e1->e2 \n\t enter -1, if e2->e1 \n\t else enter 0 \n\n");
   for (i = 0; i < e2; i++)
      printf("\te2%d", i + 1);
   for (i = 0; i < e1; i++)
   {
      printf("\n e1%d \t", i + 1);
      for (j = 0; j < e2; j++)
         scanf("%d", &dep[i][j]);
   }
```

```
    for (i = 0; i < e1; i++)
    {
        for (j = 0; j < e2; j++)
        {
            if (dep[i][j] == 1)
            {
                p2[j] = max1(p2[j], p1[i] + 1);
                for (k = j; k < e2; k++)
                    p2[k + 1] = p2[k] + 1;
            }
            if (dep[i][j] == -1)
            {
                p1[i] = max1(p1[i], p2[j] + 1);
                for (k = i; k < e1; k++)
                    p2[k + 1] = p1[k] + 1;
            }
        }
    }
    printf("P1 : ");
    for (i = 0; i < e1; i++)
    {
        printf("%d", p1[i]);
    }
    printf("\n P2 : ");
    for (j = 0; j < e2; j++)
        printf("%d", p2[j]);
    return 0;
}
```

Output :-

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL                    1: C/C++ Compile Run

C:\Users\Sandeep\Desktop\DS_LAB_File>cd /d "c:\Users\Sandeep\Desktop\DS_LAB_File\prg1"

c:\Users\Sandeep\Desktop\DS_LAB_File\prg1>.\"prg1.exe"
Enter the events : 3 4
Enter the dependency matrix:
         Enter 1 if e1->e2
         enter -1, if e2->e1
         else enter 0

        e21     e22     e23     e24
 e11    0 0 0 0

 e12    0 0 1 0

 e13    0 -1 0 0
P1 : 123
 P2 : 1234
c:\Users\Sandeep\Desktop\DS_LAB_File\prg1>
```

Aim :- *Write a program to simulate the Distributed Mutual Exclusion in 'C'.*

Code :-

```
#include <stdio.h>
#include <conio.h>
#include <dos.h>
#include <time.h>
void main()
{
   int cs = 0, pro = 0;
   double run = 5;
   char key = 'a';
   time_t t1, t2;
   printf("Press a key(except q) to enter a process into critical section.");
   printf(" \nPress q at any time to exit.");
   t1 = time(NULL) - 5;
   while (key != 'q')
   {
      while (!kbhit())
         if (cs != 0)
         {
            t2 = time(NULL);
            if (t2 - t1 > run)
            {
               printf("Process%d ", pro - 1);
               printf(" exits critical section.\n");
               cs = 0;
            }
         }
      key = getch();
      if (key != 'q')
```

```
    {
        if (cs != 0)
        {
            printf("Error: Another process is currently executing critical section Please wait till its execution is over.\n");
        }

        else
        {
            printf("Process %d ", pro);
            printf(" entered critical section\n");
            cs = 1;
            pro++;
            t1 = time(NULL);
        }
    }
  }
}
```

Output :-

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL                    1: C/C++ Compile Run

Microsoft Windows [Version 10.0.19042.
(c) 2020 Microsoft Corporation. All ri

C:\Users\Sandeep\Desktop\DS_LAB_File>c

c:\Users\Sandeep\Desktop\DS_LAB_File\prg2>.\"prg2.exe"
Press a key(except q) to enter a process into critical section.
Press q at any time to exit.Process 0  entered critical section
Process0  exits critical section.
Process 1  entered critical section
Process1  exits critical section.
Process 2  entered critical section
c:\Users\Sandeep\Desktop\DS_LAB_File\prg2>.\"prg2.exe"
Press a key(except q) to enter a process into critical section.
Press q at any time to exit.Process 0  entered critical section
Process0  exits critical section.
Process 1  entered critical section
Process1  exits critical section.
Process 2  entered critical section
Process 2  entered critical section
Process2  exits critical section.
Process 3  entered critical section
Process3  exits critical section.

c:\Users\Sandeep\Desktop\DS_LAB_File\prg2>
```

Aim :- *Write a Write a program to implement a distributed Chat Server using TCP sockets in 'C'.*

Code :-

TCP SERVER

```
#include <stdio.h>
#include <netdb.h>
#include <netinet/in.h>
#include <stdlib.h>
#include <string.h>
#include <sys/socket.h>
#include <sys/types.h>
#define MAX 80
#define PORT 8080
#define SA struct sockaddr
// Function designed for chat between client and server.
void func(int sockfd)
{
   char buff[MAX];
   int n;
   // infinite loop for chat
   for (;;)
   {
      bzero(buff, MAX);
      // read the message from client and copy it in buffer
      read(sockfd, buff, sizeof(buff));
      // print buffer which contains the client contents
      printf("From client: %s\t To client : ", buff);
      bzero(buff, MAX);
      n = 0;
      // copy server message in the buffer
```

```
        while ((buff[n++] = getchar()) != '\n')
            ;
        // and send that buffer to client
        write(sockfd, buff, sizeof(buff));
        // if msg contains "Exit" then server exit and chat ended.
        if (strncmp("exit", buff, 4) == 0)
        {
            printf("Server Exit...\n");
            break;
        }
    }
}
// Driver function
int main()
{
    int sockfd, connfd, len;
    struct sockaddr_in servaddr, cli;
    // socket create and verification
    sockfd = socket(AF_INET, SOCK_STREAM, 0);
    if (sockfd == -1)
    {
        printf("socket creation failed...\n");
        exit(0);
    }
    else
        printf("Socket successfully created..\n");
    bzero(&servaddr, sizeof(servaddr));
    // assign IP, PORT
    servaddr.sin_family = AF_INET;
    servaddr.sin_addr.s_addr = htonl(INADDR_ANY);
    servaddr.sin_port = htons(PORT);
```

```
    // Binding newly created socket to given IP and verification
    if ((bind(sockfd, (SA *)&servaddr, sizeof(servaddr))) != 0)
    {
        printf("socket bind failed...\n");
        exit(0);
    }
    else
        printf("Socket successfully binded..\n");
    // Now server is ready to listen and verification
    if ((listen(sockfd, 5)) != 0)
    {
        printf("Listen failed...\n");
        exit(0);
    }
    else
        printf("Server listening..\n");
    len = sizeof(cli);
    // Accept the data packet from client and verification
    connfd = accept(sockfd, (SA *)&cli, &len);
    if (connfd < 0)
    {
        printf("server acccept failed...\n");
        exit(0);
    }
    else
        printf("server acccept the client...\n");
    // Function for chatting between client and server
    func(connfd);
    // After chatting close the socket
    close(sockfd);
}  sfdffdf
```

TCP CLIENT

```
#include <netdb.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/socket.h>
#define MAX 80
#define PORT 8080
#define SA struct sockaddr
void func(int sockfd)
{
    char buff[MAX];
    int n;
    for (;;)
    {
        bzero(buff, sizeof(buff));
        printf("Enter the string : ");
        n = 0;
        while ((buff[n++] = getchar()) != '\n')
            ;
        write(sockfd, buff, sizeof(buff));
        bzero(buff, sizeof(buff));
        read(sockfd, buff, sizeof(buff));
        printf("From Server : %s", buff);
        if ((strncmp(buff, "exit", 4)) == 0)
        {
            printf("Client Exit...\n");
            break;
        }
```

```
    }
}
int main()
{
    int sockfd, connfd;
    struct sockaddr_in servaddr, cli;
    // socket create and varification
    sockfd = socket(AF_INET, SOCK_STREAM, 0);
    if (sockfd == -1)
    {
        printf("socket creation failed...\n");
        exit(0);
    }
    else
        printf("Socket successfully created..\n");
    bzero(&servaddr, sizeof(servaddr));
    // assign IP, PORT
    servaddr.sin_family = AF_INET;
    servaddr.sin_addr.s_addr = inet_addr("127.0.0.1");
    servaddr.sin_port = htons(PORT);
    // connect the client socket to server socket
    if (connect(sockfd, (SA *)&servaddr, sizeof(servaddr)) != 0)
    {
        printf("connection with the server failed...\n");
        exit(0);
    }
    else
        printf("connected to the server..\n");
    // function for chat
```

```
    func(sockfd);
    // close the socket
    close(sockfd);
}
```

Output :-

TCP SERVER

```
abhishek@abhishek-VirtualBox: ~/Desktop/DS
abhishek@abhishek-VirtualBox:~$ cd "Desktop/DS"
abhishek@abhishek-VirtualBox:~/Desktop/DS$ gcc chatser.c -o chatser
chatser.c: In function 'func':
chatser.c:22:3: warning: implicit declaration of function 'read'; did you mean '
fread'? [-Wimplicit-function-declaration]
   22 |   read(sockfd, buff, sizeof(buff));
      |   ^~~~
      |   fread
chatser.c:32:3: warning: implicit declaration of function 'write'; did you mean
'fwrite'? [-Wimplicit-function-declaration]
   32 |   write(sockfd, buff, sizeof(buff));
      |   ^~~~~
      |   fwrite
chatser.c: In function 'main':
chatser.c:93:2: warning: implicit declaration of function 'close'; did you mean
'pclose'? [-Wimplicit-function-declaration]
   93 |  close(sockfd);
      |  ^~~~~
      |  pclose
abhishek@abhishek-VirtualBox:~/Desktop/DS$ ./chatser
Socket successfully created..
Socket successfully binded..
Server listening..
server acccept the client...
From client: Hii,Abhishek here
        To client : hello Abhishek, Server here
From client: 60
        To client :
```

TCP CLIENT

```
abhishek@abhishek-VirtualBox: ~/Desktop/DS
abhishek@abhishek-VirtualBox:~$ cd "Desktop/DS"
abhishek@abhishek-VirtualBox:~/Desktop/DS$ gcc chatcli -o chatcli
gcc: error: chatcli: No such file or directory
gcc: fatal error: no input files
compilation terminated.
abhishek@abhishek-VirtualBox:~/Desktop/DS$ gcc chatcli.c -o chatcli
chatcli.c: In function 'func':
chatcli.c:19:3: warning: implicit declaration of function 'write'; did you mean
'fwrite'? [-Wimplicit-function-declaration]
   19 |   write(sockfd, buff, sizeof(buff));
      |   ^~~~~
      |   fwrite
chatcli.c:21:3: warning: implicit declaration of function 'read'; did you mean '
fread'? [-Wimplicit-function-declaration]
   21 |   read(sockfd, buff, sizeof(buff));
      |   ^~~~
      |   fread
chatcli.c: In function 'main':
chatcli.c:47:29: warning: implicit declaration of function 'inet_addr' [-Wimplic
it-function-declaration]
   47 |   servaddr.sin_addr.s_addr = inet_addr("127.0.0.1");
      |                               ^~~~~~~~
chatcli.c:62:2: warning: implicit declaration of function 'close'; did you mean
'pclose'? [-Wimplicit-function-declaration]
   62 |  close(sockfd);
      |  ^~~~~
      |  pclose
abhishek@abhishek-VirtualBox:~/Desktop/DS$ ./chatcli
Socket successfully created..
connected to the server..
Enter the string : Hii,Abhishek here
From Server : hello Abhishek, Server here
Enter the string : 60
```

Aim :- *Write a program on RPC mechanism for a file transfer across a network in 'C'.*

Code :-

CLIENT SIDE

```
#include "transfer.h"
#include <time.h>
   void
   transfer_1(char *host, char *filetotransf)
{
   CLIENT *clnt;
   int *result_1;
   file transf_1_arg;
   FILE *ofile;
   long long int total = 0;
   clnt = clnt_create(host, TRANSFER, TRANSFER_1, "tcp");
   if (clnt == NULL)
   {
      clnt_pcreateerror(host);
      exit(1);
   }
   ofile = fopen(filetotransf, "rb");
   if (ofile == NULL)
   {
      printf("File not found.\n");
      exit(1);
   }
   printf("Sending file %s.\n", filetotransf);
   strcpy(transf_1_arg.name, filetotransf);
   clock_t begin = clock();
   while (1)
```

```
    {
        transf_1_arg.nbytes = fread(transf_1_arg.data, 1, MAXLEN, ofile);
        total += transf_1_arg.nbytes;
        //printf("\r%lld bytes of %s sent to server.", total, transf_1_arg.name);
        result_1 = transf_1(&transf_1_arg, clnt);
        if (result_1 == (int *)NULL)
        {
            clnt_perror(clnt, "call failed");
        }
        if (transf_1_arg.nbytes < MAXLEN)
        {
            printf("\nUpload finished.\n");
            break;
        }
    }
    clock_t end = clock();
    double upload_time = (double)(end - begin) / CLOCKS_PER_SEC;
    printf("Upload time: %lf\n", upload_time);
    clnt_destroy(clnt);
    fclose(ofile);
}
int main(int argc, char *argv[])
{
    char *host;
    char *filetotransf;
    if (argc < 3)
    {
        printf("usage: %s <server_host> <file>\n", argv[0]);
        exit(1);
    }
    host = argv[1];
```

```
    filetotransf = argv[2];
    transfer_1(host, filetotransf);
    exit(0);
}
```

<u>SERVER SIDE</u>

```
#include "transfer.h"
    char opened_file[MAXLEN];
FILE *ofile;
long long int total = 0;
int *transf_1_svc(file *argp, struct svc_req *rqstp)
{
    static int result;
    static char tempName[MAXLEN];
    strcpy(tempName, "uploaded_");
    strcat(tempName, argp->name);
    strcpy(argp->name, tempName);
    total += argp->nbytes;
    if (strcmp(opened_file, "") == 0 && ofile == NULL)
    {
        printf("Receiving new file %s.\n", argp->name);
        strcpy(opened_file, argp->name);
        ofile = fopen(argp->name, "ab+");
    }
    if (strcmp(opened_file, argp->name) == 0)
    {
        //printf("\r%lld bytes of file %s were received.", total, argp->name);
        fflush(stdout);
        fwrite(argp->data, 1, argp->nbytes, ofile);
        if (argp->nbytes < MAXLEN)
        {
            printf("\nFinished receiving %s.\n", argp->name);
```

```
        total = 0;
        fclose(ofile);
        ofile = NULL;
        strcpy(opened_file, "");
      }
    }
    return &result;
}
```

Output :-

SERVER SIDE

```
Socket file descriptor 3 received

Successfully binded!

Waiting for file name...

File Name Received: dm.txt

File Successfully opened!

Waiting for file name...

File Name Received: /home/dmayank/Documents/dm.txt

File Successfully opened!
```

CLIENT SIDE

```
Socket file descriptor 3 received

Please enter file name to receive:
dm.txt

---------Data Received---------
30
-------------------------------

Please enter file name to receive:
/home/dmayank/Documents/dm.txt

---------Data Received---------
30
-------------------------------
```

Aim :- *Write a program to implement Java RMI mechanism for accessing methods of remote systems.*

Code :-

AddClient.java

```
import java.rmi.*;


public class AddClient
{
public static void main(String args[])
{
try
{
String addServerURL="rmi://"+ args[0] + "/AddServer";
AddServerIntf addServerIntf =
(AddServerIntf)Naming.lookup(addServerURL);
System.out.println("the first no is:" + args[1]);
double d1=Double.valueOf(args[1]).doubleValue();
System.out.println("the second no is:" + args[2]);
double d2=Double.valueOf(args[2]).doubleValue();
System.out.println("Sum = " + addServerIntf.add(d1,d2));
}
catch(Exception e)
{
System.out.println("Exception:" +e);
}
}
}
```

AddServer.java

```
import java.net.*;
import java.rmi.*;
```

```
public class AddServer
{
public static void main(String args[]){
try
{
AddServerImpl addServerImpl = new AddServerImpl();
Naming.rebind("AddServer", addServerImpl);
}
catch(Exception e){
System.out.println("Exception:" +e);
}
}
}AddServerImpl.java
import java.rmi.*;
import java.rmi.server.*;

public class AddServerImpl extends UnicastRemoteObject implements
AddServerIntf
{
public AddServerImpl() throws RemoteException
{
}
public double add(double d1,double d2) throws RemoteException
{
return d1+d2;
}
}AddServerIntf.java
import java.rmi.*;

public interface AddServerIntf extends Remote {
```

```
    double add(double d1, double d2) throws RemoteException;
}
```

Output :-

Output

```
// when arguments are passed as 35 and 16

Sum = 51
```

Aim :- *Write a Write a program to simulate Balanced Window Protocol in 'C'.*

Code :-

```
#include <stdio.h>
int main()
{
   int w, i, f, frames[50];
   printf("Enter window size: ");
   scanf("%d", &w);
   printf("\nEnter number of frames to transmit: ");
   scanf("%d", &f);
   printf("\nEnter %d frames: ", f);
   for (i = 1; i <= f; i++)
      scanf("%d", &frames[i]);
   printf("\nWith sliding window protocol the frames will be sent in the following manner
(assuming no corruption of frames)\n\n");
   printf("After sending %d frames at each stage sender waits for acknowledgement sent by
the receiver\n\n", w);
   for (i = 1; i <= f; i++)
   {
      if (i % w == 0)
      {
         printf("%d\n", frames[i]);
         printf("Acknowledgement of above frames sent is received by sender\n\n");
      }
      else
         printf("%d ", frames[i]);
   }
   if (f % w != 0)
      printf("\nAcknowledgement of above frames sent is received by sender\n");
   return 0;
}
```

Output :-

```
PROBLEMS 7   OUTPUT   DEBUG CONSOLE   TERMINAL                    2: C/C++ Compile Run

(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\Sandeep\Desktop\DS_LAB_File>cd /d "c:\Users\Sandeep\Desktop\DS_LAB_File\prg6"

c:\Users\Sandeep\Desktop\DS_LAB_File\prg6>.\"prg6.exe"
Enter window size: 5

Enter number of frames to transmit: 8

Enter 8 frames: 12 3 8 7 90 56 28 5

With sliding window protocol the frames will be sent in the following manner (assuming no corruption of frames)

After sending 5 frames at each stage sender waits for acknowledgement sent by the receiver

12 3 8 7 90
Acknowledgement of above frames sent is received by sender

56 28 5
Acknowledgement of above frames sent is received by sender

c:\Users\Sandeep\Desktop\DS_LAB_File\prg6>
```

Aim :- *Write a Write a program to implement COBRA mechanism using 'C++' program at one end and Java program on the other.*

Code :-

Creating the Server

```
#include <iostream.h>
#include "OB/CORBA.h"
#include <OB/Cosnaming.h>
#include "crypt.h"
#include "cryptimpl.h"
   using namespace std;
int main(int argc, char **argv)
{
   // Declare ORB and servant object
   CORBA::ORB_var orb;
   CryptographicImpl *CrypImpl = NULL;
   try
   {
      // Initialize the ORB.
      orb = CORBA::ORB_init(argc, argv);
      // Get a reference to the root POA
      CORBA::Object_var rootPOAObj = orb -
                              > resolve_initial_references("RootPOA");
      // Narrow it to the correct type
      PortableServer::POA_var rootPOA =
         PortableServer::POA::_narrow(rootPOAObj.in());
      // Create POA policies
      CORBA::PolicyList policies;
      policies.length(1);
      policies[0] = rootPOA->create_thread_policy(PortableServer::SINGLE_THREAD_MODEL);
      // Get the POA manager object
```

```
PortableServer::POAManager_var manager = rootPOA->the_POAManager();
// Create a new POA with specified policies
PortableServer::POA_var myPOA = rootPOA->create_POA("myPOA",
                                        manager, policies);
// Free policies
CORBA::ULong len = policies.length();
for (CORBA::ULong i = 0; i < len; i++)
   policies[i]->destroy();
// Get a reference to the Naming Service root_context
CORBA::Object_var rootContextObj = orb -
                              > resolve_initial_references("NameService");
// Narrow to the correct type
CosNaming::NamingContext_var nc =
   CosNaming::NamingContext::_narrow(rootContextObj.in());
// Create a reference to the servant
CrypImpl = new CryptographicImpl(orb);
// Activate object
PortableServer::ObjectId_var myObjID = myPOA->activate_object(CrypImpl);
// Get a CORBA reference with the POA through the servant
CORBA::Object_var o = myPOA->servant_to_reference(CrypImpl);
// The reference is converted to a character string
CORBA::String_var s = orb->object_to_string(o);
cout << "The IOR of the object is: " << s.in() << endl;
CosNaming::Name name;
name.length(1);
name[0].id = (const char *)"CryptographicService";
name[0].kind = (const char *)"";
// Bind the object into the name service
nc->rebind(name, o);
// Activate the POA
manager->activate();
```

```
        cout << "The server is ready. Awaiting for incoming requests..." << endl;
        // Start the ORB
        orb->run();
    }
    catch (const CORBA::Exception &e)
    {
        // Handles CORBA exceptions
        cerr << e << endl;
    }
    // Decrement reference count
    if (CrypImpl)
        CrypImpl->_remove_ref();
    // End CORBA
    if (!CORBA::is_nil(orb))
    {
        try
        {
            orb->destroy();
            cout << "Ending CORBA..." << endl;
        }
        catch (const CORBA::Exception &e)
        {
            cout << "orb->destroy() failed:" << e << endl;
            return 1;
        }
    }
    return 0;
}
```

Implementing the Client

```
#include <iostream.h>
#include <string.h>
```

```
#include "OB/CORBA.h"
#include "OB/Cosnaming.h"
#include "crypt.h"
   using namespace std;
int main(int argc, char **argv)
{
   // Declare ORB
   CORBA::ORB_var orb;
   try
   {
      // Initialize the ORB
      orb = CORBA::ORB_init(argc, argv);
      // Get a reference to the Naming Service
      CORBA::Object_var rootContextObj = orb -
                                 > resolve_initial_references("NameService");
      CosNaming::NamingContext_var nc =
         CosNaming::NamingContext::_narrow(rootContextObj.in());
      CosNaming::Name name;
      name.length(1);
      name[0].id = (const char *)"CryptographicService";
      name[0].kind = (const char *)"";
      // Invoke the root context to retrieve the object reference
      CORBA::Object_var managerObj = nc->resolve(name);
      // Narrow the previous object to obtain the correct type
      ::CaesarAlgorithm_var manager =
         ::CaesarAlgorithm::_narrow(managerObj.in());
      string info_in, exit, dummy;
      CORBA::String_var info_out;
      ::CaesarAlgorithm::charsequence_var inseq;
      unsigned long key, shift;
      try
```

```
{
    do
    {
        cout << "\nCryptographic service client" << endl;
        cout << "----------------------------" << endl;
        do
        { // Get the cryptographic key
            if (cin.fail())
            {
                cin.clear();
                cin >> dummy;
            }
            cout << "Enter encryption key: ";
            cin >> key;
        } while (cin.fail());
        do
        { // Get the shift
            if (cin.fail())
            {
                cin.clear();
                cin >> dummy;
            }
            cout << "Enter a shift: ";
            cin >> shift;
        } while (cin.fail());
        // Used for debug pourposes
        //key = 9876453;
        //shift = 938372;
        getline(cin, dummy); // Get the text to encrypt
        cout << "Enter a plain text to encrypt: ";
        getline(cin, info_in);
```

```
            // Invoke first remote method
            inseq = manager->encrypt(info_in.c_str(), key, shift);
            cout << "----------------------------------------" << endl;
            cout << "Encrypted text is: " << inseq->get_buffer() << endl;
            // Invoke second remote method
            info_out = manager->decrypt(inseq.in(), key, shift);
            cout << "Decrypted text is: " << info_out.in() << endl;
            cout << "----------------------------------------" << endl;
            cout << "Exit? (y/n): ";
            cin >> exit;
        } while (exit != "y");
        // Shutdown server message
        manager->shutdown();
    } // end of tyr2
    catch (const std::exception &std_e)
    {
        cerr << std_e.what() << endl;
    }
} //end of try1
catch (const CORBA::Exception &e)
{
    // Handles CORBA exceptions
    cerr << e << endl;
} // End CORBA
if (!CORBA::is_nil(orb))
{
    try
    {
        orb->destroy();
        cout << "Ending CORBA..." << endl;
    }
```

```
    catch (const CORBA::Exception &e)
    {
      cout << "orb->destroy failed:" << e << endl;
      return 1;
    }
  }
  return 0;
} //end of main
```

Output :-

Running the Client-server Application Once we have implemented the client and the server, it"s time to connect them. Because our demonstration client and server exchange object references via the naming service, we must ensure that the naming service (which is called nameserv in Orbacus) is running. We use some command-line options to tell the naming service the host and port on which it should listen. nameserv -OAhost localhost -OAport 8140 After this, we can start the server with a command-line option to tell it how to contact the naming service.

server -ORBInitRef NameService=corbaloc:iiop:localhost:8140/NameService

Finally we can start the client, again with a command-line option to tell it how to contact the naming service.

client -ORBInitRef NameService=corbaloc:iiop:localhost:8140/NameService