Name of the Subject: DISTRIBUTED COMPUTER Subject Code: IT-717

Seat No: IT076 Student ID: 18ITUBN116 Branch/Sem: IT-VII

Q4 C

① STTPS Scientific Serv. java.

```
@WebService (ServiceName = "STTPScientific WSDL Servic",
    PortName = "STTPScientific WSDL Port", endpointInterface =
    "org.netbeans.jree.wsdl. STTPS Scientific WSDL PortType")

public clas STTPScientific Servic implements
                    STTPScientific WSDL PortType
{
    Public double Sin (double Sin RequestPart)
    {
        Math.Sin (Sin RequestPart * Math.PI / 180);
    }
}
```

② addition.java.

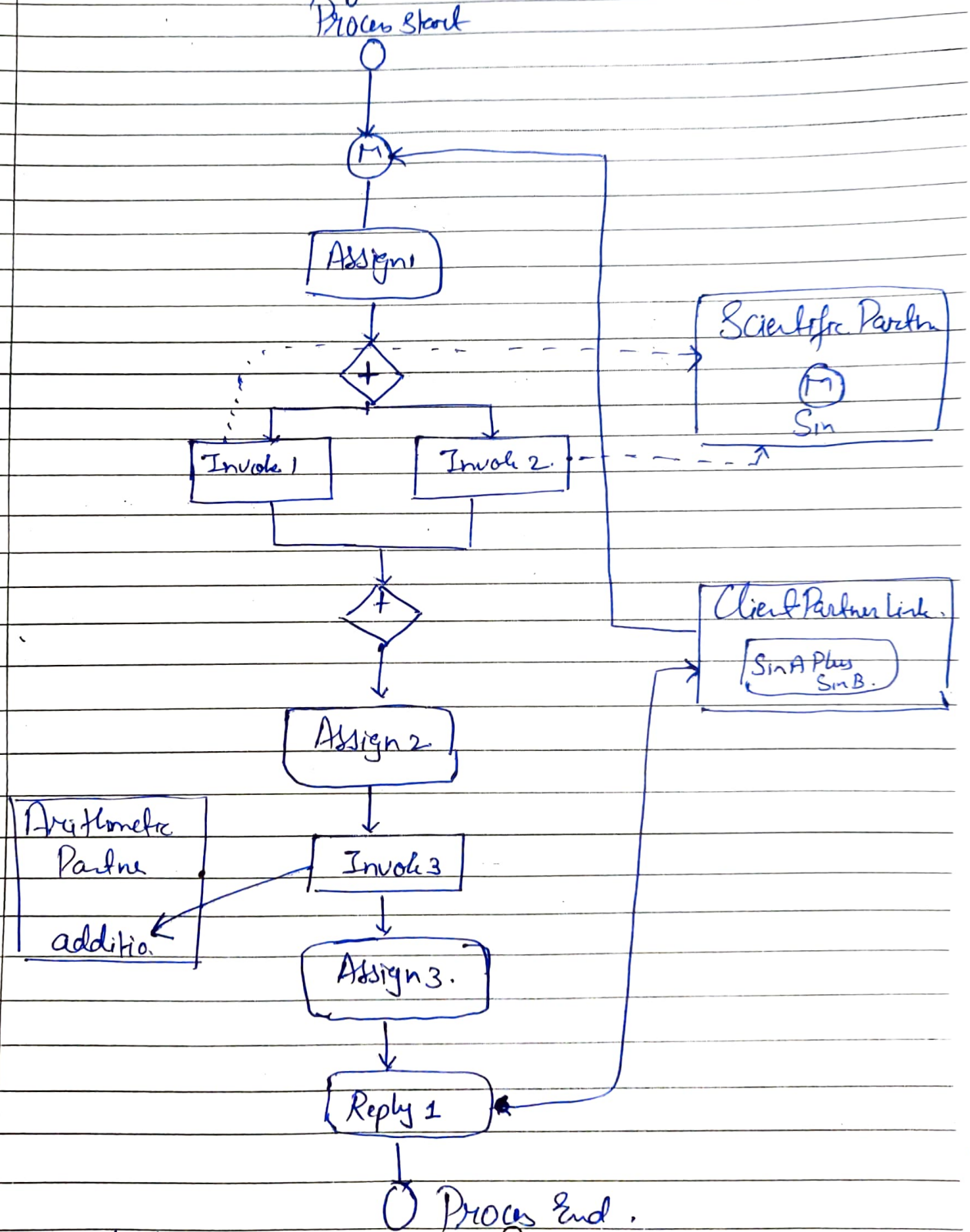
```
package Src:

@WebService ()
public clas addition
{
    @WebMethod (Operation Name = "addition")
    pubic double addition (@WebParam (name = "Param1")
        double param1, @WebParam (name = "Param 2")
        double param 2)
    {
        return param 1 + param 2;
    }
}
```

Section – II Date: 24/11/2021 Signature of Student: Page No: 01

Name of the Subject: DISTRIBUTED COMPUTER Subject Code: IT-717

Seat No: IT076 Student ID: 18ITUBN116 Branch/Sem: IT-VII

(ii) Draw BPEL diagram.



(1) We Send the input to BPEL process using client partner by receiving activity that receive activity assign variable

(2) Then the Addition Palm link value from the user to send & then addition process is done by the Invoke activity.

(3) Then with the Trigo Partner link the Trigo Calculate is done & it is send to to reply activity.

(4) The reply activity get the answer & send it to Client Partner Link.

(iii) Step to Create Interface Using Code First Approach

(1) Create Web Application — give name addition
— give it name STTArithServer

(2) Now, Select new & add web service & the Provide Service Name & Package Name.

(3) Go to Design View, & then add the operation for addition & also Provide operation Name & Return Type.

(4) Then go to the Source View of Web Service & Modify the code of addition operation clean & build the project.

(5) Select Project
└ Web Service
└ Arithmetic Ser...
└ TestWebSer...

Name of the Subject: DC Subject Code: IT-717

Seat No: IT076 Student ID: 18ITUBN116 Branch/Sem: IT-VII

Q4 b

@ Path

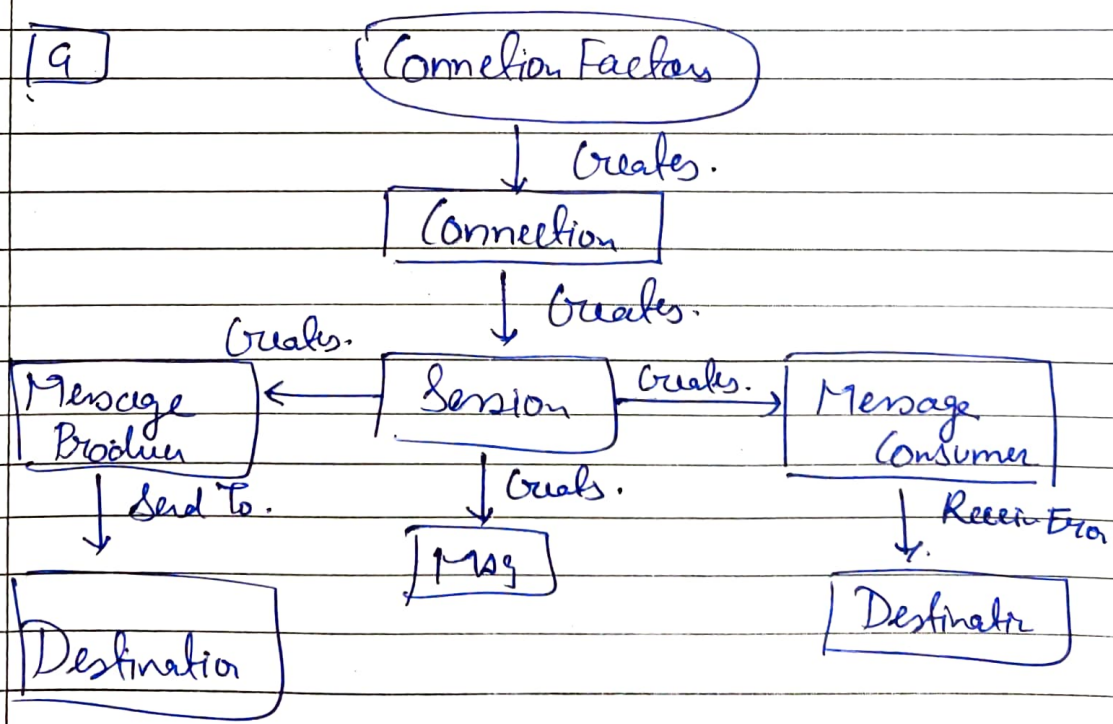
The @Path annotation's value is a relative URI path indicating where the Java class will be hosted.

@ PathParam

The @PathParam annotation is a type of parameter that you can extract for use in your resource class.

URI path parameters are extracted from request URI, & the parameter name corresponds to the URI path template variable name specified in the @Path class-level annotation.

Q4 a



Name of the Subject: DC Subject Code: IT-717

Seat No: IT076 Student ID: 18ITUBN116 Branch/Sem: IT-VII

JMS - Administrated object
- Connections
- Sessions
- Message Producer
- Message Consumer
- Message.

Connection Factory - Provides the connection btw JMS Client & Service provider.

Destination - JMS Client to target the message & receive message from destination.

Connection - JMS provider such as WebSphere, Active MQ. To create Connection with client & defines these provider resources virtually outside JVM.

Session - You can create messages producer, message Consumer & message by using Session.

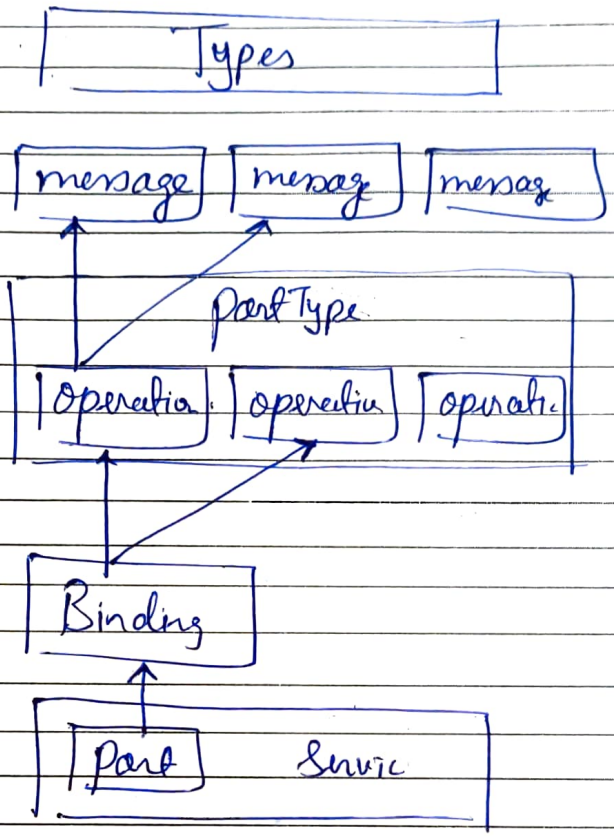
Message Producer - Generated by a Session which send message to the destination by implementing MessageProducer interface.

Message Consumer - Generated by a Session, which receive message from the destination by implementing the MessageConsumer.

Message - Includes the data, which will be exchanged btw JMS Clients to design of a JMS Application.

Section - II Date: 24/11/2021 Signature of Student: Page No: 05

Q3 d. WSDL.




- Definition - Must be the root element, it should define the name of the servics.
  - Declare the namespace used in the document.
- Types - Describe all the data types used by the client & Server & can be omitted of only Simple data types are used.

Name of the Subject: DC Subject Code: IT-717

Seat No: IT076 Student ID: 18ITUBN116 Branch/Sem: IT-VII

- Message - Define the name of the request or response message
  - Define also the message part elements.

- PortType - Defines the combination of message elements to form a complete one-way or round trip operation.

- Binding
  - Provide specific details on how a portType operation will actually be transmitted over the wire.
  - SOAP specific information can be defined here. WSDL includes built-in extension for defining SOAP services.

- Service
  - Define the address for invoking the specific service

- documentation
  - Provide human-readable documentation

- import
  - Enable a more modular WSDL document.

Section – II Date: 24/11/2021 Signature of Student: Page No: 07

Name of the Subject: DC Subject Code: IT-717

Seat No: IT076 Student ID: 18ITUBN116 Branch/Sem: IT-VII

Q3 a.

TCP Client-Server Program UNIX close() Function

- It is used to close a socket & terminate a TCP connection.
- Return 0 if ok, -1 on error.
- In Concurrent Server, what if the parent does not close connection.
- close() marks socket as closed & return immediately, - Sockfd is no longer usable.
- TCP continues to try sending unsent data.

TCP Client-Server Program UNIX close()

- If you want to close the connection normally, Shutdown the connection with SHUT-WR & wait until you receive a 0 size data, & then close the socket
- With Shutdown, you will still able to receive pending data the peer already sent.

Name of the Subject: DC Subject Code: IT-717

Seat No: IT076 Student ID: 18ITUBN116 Branch/Sem: IT-VII

Q3 (a) b

Server.c.

```
#include <string.h>
#include <sys/ioctl.h>
#include <stdlib.h>

int main()
{
    int s-b, b, c_d;
    struct fname[50], op[1000];
    struct sockaddr_in caddr, saddr;
    FILE *fp;

    socklen_t clen = sizeof(caddr);
    sd = socket(AF_INET, SOCK_STREAM, 0);

    if (sd != -1)
        printf("Socket created");
    else
        printf("Not created");

    saddr.sin_family = AF_INET;
    saddr.sin_port = htons(7069);
    saddr.sin_addr.s_addr = htonl(INADDR_ANY);

    b = bind(sd, (struct sockaddr*) &saddr, sizeof(saddr));
    if (b == 0)
        printf("binded Success");
    else
        printf("binded failed");
    listen(sd, 5);
```

Section – II Date: 24/11/2021 Signature of Student: Page No: 09.

```
cd = accept (sd, (struct sockaddr *)& caddr, &clen);
recv (cd, fname, sizeof (fname), 0);
fp = open (fname, "w");
fwrite (op, strlen (op), 1, fp);
printf (" File Transferred");
close (fd);
close (cd); fclose (fp);

return 0;
}
```

3.

## (2) Client Side

```
#include <string.h>
#include <sys/ioctl.h>
#include <socket.h>
#include <netinet/in.h>

int main()
{
    int s_d, c, s;
    char fname[50], sip[25], op[1000];
    struct sockaddr_in caddr;
    struct hostent *h;
    FILE *fp;
    printf ("Enter Server IP");
    scanf ("%s", sip);
    h = gethostbyname (sip);
```

Name of the Subject: DC Subject Code: IT-717

Seat No: IT076 Student ID: 18ITUBN116 Branch/Sem: IT-VII

```
sd = socket(AF_INET, SOCK_STREAM, 0);
if(sd != 1)
    printf("Socket Created");
else
    printf("Not Created");
caddr.sin_port = htons(7069);
caddr.sin_addr = *((struct in_addr*)h->h_addr);
c = connect(sd, (struct sockaddr*)&caddr, sizeof
                                    (caddr));
if(c==0)
    printf("Connect Serv");
else
    printf("Failed");
printf("File Name");
scanf("%s", fname);
send(sd, fname, sizeof(fname), 0);
fp = fopen(fname, "r");
fseek(fp, 1000, 1, fp);
send(sd, op, sizeof(op), 0);
fclose(fp);
close(sd);
return 0;
}.
```

Section – II Date: 24/11/2021 Signature of Student: Page No: 4

Name of the Subject: DC Subject Code: IT-717

Seat No: IT076 Student ID: 18ITUBN116 Branch/Sem: IT-VII

Output

- Input for Client Side

Server IP - 127.0.0.1

Connect Server.

File Name Dishant.txt
Dishant.txt

Dishant Modh.
IT076
Sem - 7

- Server Side.

Socket Created.
Binded Success.
File name: D.txt.

Dishant Modh.
IT076
Sem-7