

- 1.1 Addition Server: Implement multi-threaded client/server Process communication using RMI for Addition of Numbers.
- 1.2 Multiplication Server: Implement multi-threaded client/server Process communication using RMI for Multiplication of Numbers.
- 1.3 Division Server: Implement multi-threaded client/server Process communication using RMI for Division of Numbers.
- 1.4 Subtract Server: Implement multi-threaded client/server Process communication using RMI for Subtract of Numbers.
- 1.5 Power Calculation: Implement multi-threaded client/server Process communication using RMI for 2's Power of Given Number.
- 1.6 Celsius to Fahrenheit Conversion Server: Implement multi-threaded client/server Process communication using RMI for Celsius to Fahrenheit Conversion Server.
- 1.7 Miles to Kilometer Conversion Server: Implement multi-threaded client/server Process communication using RMI for Miles to Kilometer Conversion Server.
- 1.8 Echo Server: Implement multi-threaded client/server Process communication using RMI for Printing Name Appending to Hello.
- 1.9 Compare 2 Strings: Implement multi-threaded client/server Process communication using RMI for Comparing 2 Strings and Return lexicographically Largest string.

1.10 Count Vowels: Implement multi-threaded client/server Process communication using RMI for Counting Vowels Present in Word.

1.11 Factorial of Number: Implement multi-threaded client/server Process communication using RMI for Find factorial of Number.

2.1 Design a distributed application using MPI for computation where root process has an array of elements equal to the size of processors which is divided to the worker processes which calculates and displays the intermediate sums calculated at different processors.

2.2 Design a distributed application using MPI for computation where root process has an array of elements equal to the size of processors which is divided to the worker processes which calculates and displays the intermediate multiplication calculated at different processors.

3.1 Develop String Reversing distributed application using CORBA to demonstrate object brokering.

3.2 Develop Changing Case of String to Uppercasing distributed application using CORBA to demonstrate object brokering.

4.1 Implement Berkeley algorithm for clock synchronization.

4.2 Implement clock synchronization with Berkeley algorithm.

4.3 Implement Berkeley algorithm for clock synchronization.

5.1 Implement token ring based mutual exclusion algorithm.

5.2 Implement Mutual Exclusion based Token Ring Algorithm.

5.3 Implement token ring based mutual exclusion algorithm

6.1 Perform Leader Election with Any Algorithm.

6.2 Implement Ring algorithm for leader election.

6.3 Perform Leader Election with Any Algorithm.

6.4 Implement Bully algorithm for leader election.

7.1 Create a Calculator web service and consume that web service.

7.2 Create a Simple Interest Calculator web service and consume that web service.

7.3 Create a web service which takes User's Name as Input and Display Hello User_Name Entered by User and Consume that web service.

7.4 Create a web service which takes Fahrenheit as Input and Display Celsius as Output and Consume that web service.

7.5 Create a web service which takes Miles as Input and Display Kilometer as Output and Consume that web service.