 **Download MPJ Express**

* Go to: <http://mpj-express.org/>
* Click **Downloads** > download latest version (e.g., mpj-v0.44.zip)

 **Extract the zip file**

* Put it somewhere, e.g., C:\mpj-v0.44\

 **Set Environment Variables**

* Open Environment Variables (Edit the system environment variables)
* Add to PATH:

makefile

CopyEdit

C:\mpj-v0.44\bin

* Create a new variable:

ini

CopyEdit

MPJ\_HOME = C:\mpj-v0.44

import mpi.\*;

public class DistributedReciprocal {

public static void main(String[] args) throws MPIException {

// Initialize MPI

MPI.Init(args);

int rank = MPI.COMM\_WORLD.Rank(); // Get process rank

int size = MPI.COMM\_WORLD.Size(); // Get total number of processes

// Declare the array in the root process

int[] numbers = null;

if (rank == 0) {

// Root process creates an array of size equal to number of processes

numbers = new int[size];

System.out.println("Generated numbers:");

for (int i = 0; i < size; i++) {

numbers[i] = i + 1; // Assign values 1 to size to the array

System.out.print(numbers[i] + " ");

}

System.out.println();

}

// Array to hold local numbers for each process (size 1 for each process)

int[] localNumbers = new int[1];

// Scatter the numbers to all processes

MPI.COMM\_WORLD.Scatter(numbers, 0, 1, MPI.INT, localNumbers, 0, 1, MPI.INT, 0);

// Each process calculates the reciprocal of the number it received

double[] localReciprocal = new double[1];

localReciprocal[0] = 1.0 / localNumbers[0];

// Each process prints the reciprocal of its number

System.out.println("Process " + rank + " received " + localNumbers[0] + " and calculated reciprocal: " + localReciprocal[0]);

// Root process will gather all the reciprocals

double[] gatheredReciprocals = null;

if (rank == 0) {

gatheredReciprocals = new double[size];

}

// Gather the reciprocals from all processes

MPI.COMM\_WORLD.Gather(localReciprocal, 0, 1, MPI.DOUBLE, gatheredReciprocals, 0, 1, MPI.DOUBLE, 0);

// Root process prints the resulting array of reciprocals

if (rank == 0) {

System.out.println("Resulting array of reciprocals:");

for (int i = 0; i < size; i++) {

System.out.print(gatheredReciprocals[i] + " ");

}

System.out.println();

}

// Finalize MPI

MPI.Finalize();

}

}

Powershell

Compile:

javac -cp ".;$env:MPJ\_HOME\lib\mpj.jar" DistributedRecipocal.java

Run:

mpjrun.bat -np 4 DistributedRecipocal