import mpi.MPI;

public class ScatterGather {

public static void main(String args[]){

//Initialize MPI execution environment

MPI.Init(args);

//Get the id of the process

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

//total number of processes is stored in size

int size = MPI.COMM\_WORLD.Size();

int root=0;

//array which will be filled with data by root process

int sendbuf[]=null;

sendbuf= new int[size];

//creates data to be scattered

if(rank==root){

sendbuf[0] = 10;

sendbuf[1] = 20;

sendbuf[2] = 30;

sendbuf[3] = 40;

//print current process number

System.out.print("Processor "+rank+" has data: ");

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

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

}

System.out.println();

}

//collect data in recvbuf

int recvbuf[] = new int[1];

//following are the args of Scatter method

//send, offset, chunk\_count, chunk\_data\_type, recv, offset, chunk\_count, chunk\_data\_type, root\_process\_id

MPI.COMM\_WORLD.Scatter(sendbuf, 0, 1, MPI.INT, recvbuf, 0, 1, MPI.INT, root);

System.out.println("Processor "+rank+" has data: "+recvbuf[0]);

System.out.println("Processor "+rank+" is doubling the data");

recvbuf[0]=recvbuf[0]\*2;

//following are the args of Gather method

//Object sendbuf, int sendoffset, int sendcount, Datatype sendtype,

//Object recvbuf, int recvoffset, int recvcount, Datatype recvtype,

//int root)

MPI.COMM\_WORLD.Gather(recvbuf, 0, 1, MPI.INT, sendbuf, 0, 1, MPI.INT, root);

//display the gathered result

if(rank==root){

System.out.println("Process 0 has data: ");

for(int i=0;i<4;i++){

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

}

}

//Terminate MPI execution environment

MPI.Finalize();

}

}