```
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();

 }

}

/* OUTPUT :

sayli@sayli:~/Desktop/CL9/Ass2$ export MPJ_HOME=/home/jayshree/Desktop/CL9/Ass2/mpj-v0_44
sayli@sayli:~/Desktop/CL9/Ass2$ javac -cp $MPJ_HOME/lib/mpj.jar ScatterGather.java
sayli@sayli:~/Desktop/CL9/Ass2$ $MPJ_HOME/bin/mpjrun.sh -np 4 ScatterGather

MPJ Express (0.44) is started in the multicore configuration

 Processor 0 has data: 10 20 30 40

Processor 0 has data: 10

Processor 0 is doubling the data

Processor 1 has data: 20

Processor 2 has data: 30

Processor 3 has data: 40

Processor 3 is doubling the data

Processor 2 is doubling the data

Processor 1 is doubling the data

Process 0 has data: 20 40 60 80

sayli@sayli:~/Desktop/CL9/Ass2$
```