Client

```
import java.io.*; // BufferReader, PrintWriter
import java.net.*;
class RpcClient
{
        public static void main(String[] args) throws Exception
        {
                Socket sock = new Socket("127.0.0.1", 3000);
/*** InputStreamReader class (java.io.InputStreamReader) is intended to wrap an InputStream, thereby turning the byte based input stream into a character based Reader.
BufferedReader class is used to read the text from a character-based input stream. It can be used to read data line by line by readLine() method ***/
                BufferedReader keyRead = new BufferedReader(new InputStreamReader(System.in));

    /********** Stream is sequence of data, InputStream is used for reading, OutputStream for writing  **********/
    /********** getOutputStream is input to the process, getInputStream is output read from the process **********/
                OutputStream ostream = sock.getOutputStream();

/*** PrintWriter  Prints formatted representations of objects to a text-output stream ***/
                PrintWriter pwrite = new PrintWriter(ostream, true);

                InputStream istream = sock.getInputStream();

/*** BufferedReader  reads text from a character-input stream, buffering characters so as to provide for the efficient reading of characters, arrays, and lines ***/
                BufferedReader receiveRead = new BufferedReader(new InputStreamReader(istream));

                System.out.println("*** You are going to add three numbers ***\n");
                String receiveMessage, sendMessage,temp;

                System.out.println("Enter 1st parameter :");
                sendMessage = keyRead.readLine();
                pwrite.println(sendMessage);

                System.out.println("Enter 2nd parameter : ");
                sendMessage = keyRead.readLine();
                pwrite.println(sendMessage);

                System.out.println("Enter 3rd parameter : ");
                sendMessage = keyRead.readLine();
                pwrite.println(sendMessage);
 /***  flush() is used to empty the buffer data between the server and the client. ***/
                System.out.flush();

                if((receiveMessage = receiveRead.readLine()) != null)
                        System.out.println(receiveMessage);

        }
}
```

Server

```
import java.io.*;
import java.net.*;
class RpcServer
```

```
{
	public static void main(String[] args) throws Exception
	{
		ServerSocket sersock = new ServerSocket(3000);
		System.out.println("Server ready");
		Socket sock = sersock.accept( );

		OutputStream ostream = sock.getOutputStream();
		PrintWriter pwrite = new PrintWriter(ostream, true);

		InputStream istream = sock.getInputStream();
		BufferedReader receiveRead = new BufferedReader(new InputStreamReader(istream));

		String receiveMessage, sendMessage,fun;
		int a,b,c,d;

		a = Integer.parseInt(receiveRead.readLine());
		System.out.println("1st Parameter : "+a);

		b = Integer.parseInt(receiveRead.readLine());
		System.out.println("2nd Parameter : "+b);

		c = Integer.parseInt(receiveRead.readLine());
		System.out.println("3rd Parameter : "+c);

		d=a+b+c;
		System.out.println("Addition = "+d);
		pwrite.println("Addition = "+d);

		System.out.flush();
	}
}
```

OUTPUT