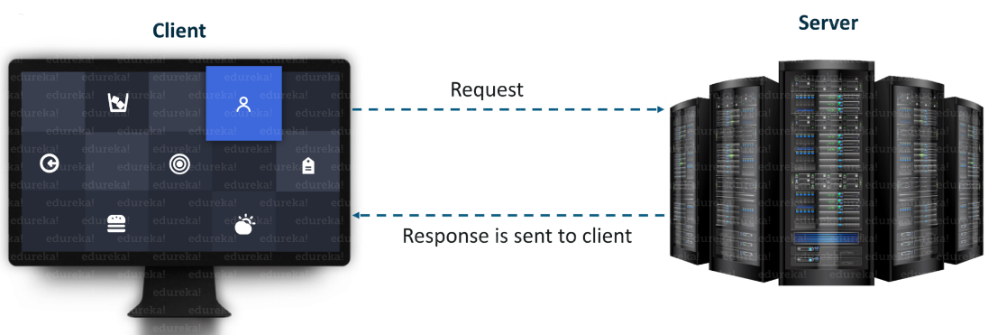
**EXP 2 STUDY OF BASIC FUNCTIONS OF SOCKET PROGRAMMING**

**What is Socket Programming in Java?**

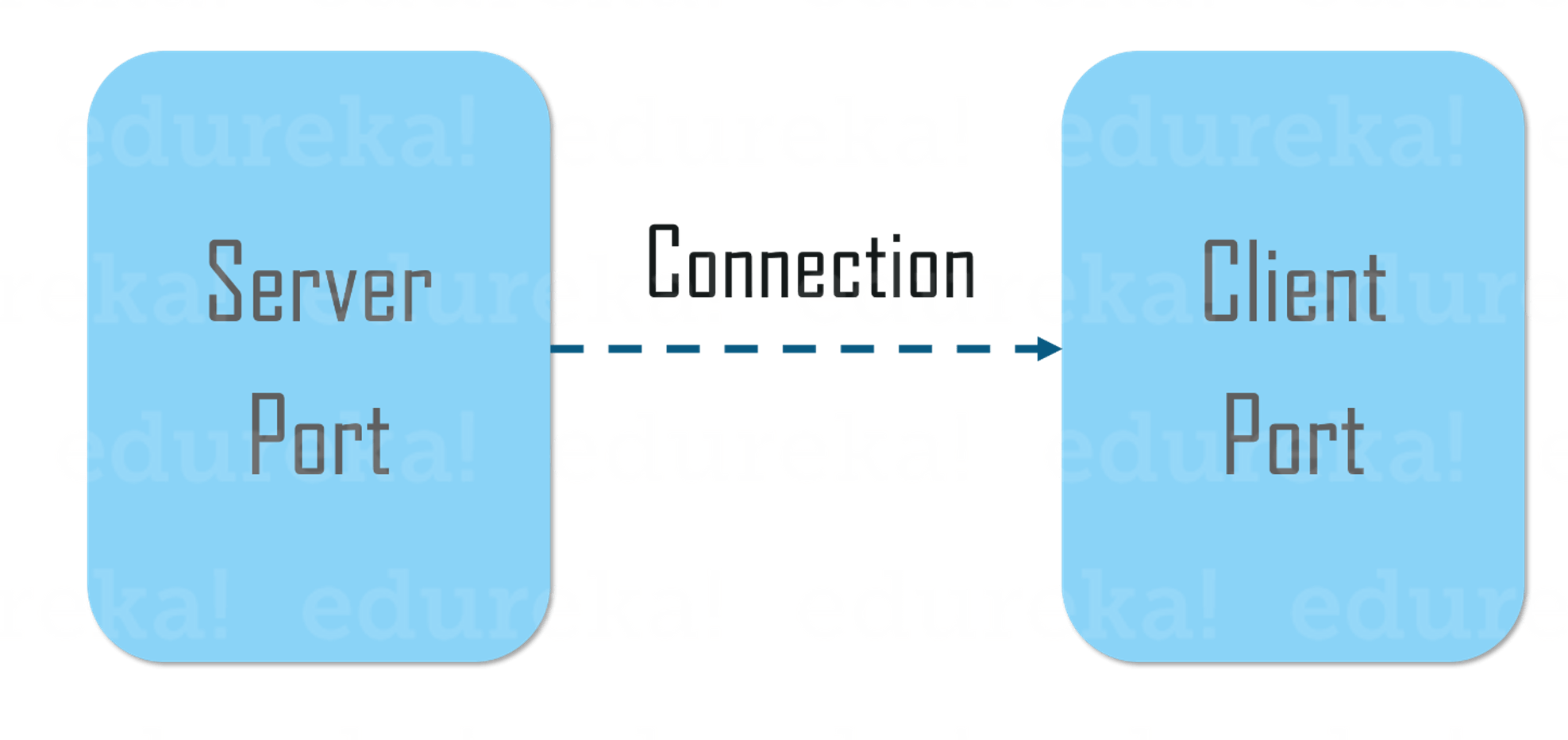
*Socket programming* is a way of connecting two nodes on a network to communicate with each other. One ***socket***(node) listens on a particular port at an IP, while other *socket*reaches out to the other in order to form a connection.



The server forms the listener *socket while* the client reaches out to the server. Socket and Server Socket [classes](https://www.edureka.co/blog/java-tutorial/#obj) are used for connection-oriented socket programming.

## ****What is a Socket in Java?****

A **socket**in [Java](https://www.edureka.co/blog/java-tutorial/) is one endpoint of a two-way communication link between two programs running on the network. A **socket** is bound to a port number so that the TCP layer can identify the application that data is destined to be sent to.



An endpoint is a combination of an IP address and a port number. The package in the Java platform provides a class, Socket that implements one side of a two-way connection between your Java program and another program on the network. The class sits on top of a platform-dependent implementation, hiding the details of any particular system from your Java program. By using the class instead of relying on native code, your [Java programs](https://www.edureka.co/blog/java-programs/) can communicate over the network in a platform-independent fashion.

## ****Client Side Programming****

In the case of client-side programming, the client will first wait for the server to start. Once the server is up and running, it will send the requests to the server. After that, the client will wait for the response from the server. So, this is the whole logic of client and server communication. Now let’s understand the client side and server side programming in detail.

In order to initiate a clients request, you need to follow the below-mentioned steps:

**1. Establish a Connection**

The very first step is to establish a socket connection. A socket connection implies that the two machines have information about each other’s network location (IP Address) and TCP port.

You can create a Socket with the help of a below statement:

Socket socket = new Socket(“127.0.0.1”, 5000)

* Here, the first argument represents the **IP address of Server**.
* The second argument represents the **TCP Port**. (It is a number that represents which application should run on a server.)

**2. Communication**

In order to communicate over a socket connection, streams are used for both input and output the data. After establishing a connection and sending the requests, you need to close the connection.

getInputStream() – method is used to get input from the end user.

getOutputStream()- method is used to send the output through the socket.

readLine()- used to read a single line of text from the console.

**3. Closing the connection**

The socket connection is closed explicitly once the message to the server is sent.

input.close()- method closes the scanner class for taking the input.

out.close()- method closes the output stream.

socket.close()- method closes the socket connection.

out.flush()- method flushes the stream, if the stream has saved any characters from the various write() methods in a buffer.

## ****Server Side Programming****

Basically, the server will instantiate its object and wait for the client request. Once the client sends the request, the server will communicate back with the response.

In order to code the server-side application, you need two sockets and they are as follows:

* A **ServerSocket** which waits for the client requests (when a client makes a new Socket())
* A plain old **socket** for communication with the client.

After this, you need to communicate with the client with the response.

**Communication**

getInputStream() – method is used to get input from the end user.

getOutputStream()- method is used to send the output through the socket.

**Close the Connection**

It is important to close the connection by closing the socket as well as input/output streams once everything is done.

## ****URL Class****

[Java](https://www.edureka.co/blog/what-is-java/) URL class mainly deals with URL(Uniform Resource Locator) which is used to identify the resources on the internet.

For Example: **https://www.java.co/blog**

Here,   https: -> Protocol  
www.java.co -> hostname  
/blog - > filename

URL Class comprises of various methods to return the URL information of a particular website.

**Various methods of Java URL Class.**

1. **getProtocol() :**Returns protocol of URL
2. **getHost() :**Returns hostname(domain name) of the specified URL
3. **getPort() :**Returns port number of the URL specified
4. **getFile() :**Returns filename of the URL