4. Network Programming with java.net Package :

Enables network communication using classes and interfaces for TCP/IP protocols.

Key Classes :-

→ **Socket** – Represents a client-side socket
→ **Server-Socket** – Used for server-side socket programming.
→ **InetAddress** – Represents IP address.
→ **URL** – Represents Uniform Resource Locator.
→ **URLConnection** – Represents a communication link b/w the application b/w application & URL

Client-Server Communication :

Client Program :-

```
import java.io.*;
import java.net.*;
public class SimpleClien {
  public static void main(String[] args) {
    try {
    Socket socket = new Socket("localhost", 5000);
    PrintWriter out = new PrintWriter(socket.getOutputStream(), tru
    out.println("Hello, Server!");
    Socket.close();
  } catch (IOException e) {
    e.printStackTrace();
  }
 }
}
```

PAGE NO. .....

Server Program :-

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

public class SimpleServer {
 public static void main (String [] args) {
  try {
   ServerSocket serverSocket = new ServerSocket (5000);
  Socket clientSocket = serverSocket.accept();
  BufferedReader in = new BufferedReader (new
  InputStreamReader (clientSocket.getInputStream()));
      System.out.println ("Client says: " + in.readLine());
      serverSocket.close();
     } catch (IOException e) {
      e.printStackTrace();
     }
    }
  }
```

5. Content and Protocol Handlers :

Content Handlers :- Manages how data from a specific content type is processed.

Protocol Handlers :- Manages communication protocols like HTTP, FTP etc.

Example –

Using URL and URL connection:-

```
Import java.net;
Import java.io;
Public class URL Example {
 Public static void main (String [] args) {
  try {
   URL url = new URL ("http://example.com");
   URL Connection conn = url.openConnection();
  BufferedReader in = new BufferedReader (new InputStream Reader);
  String InputLine;
  while ((inputLine = in.readLine()) != null){
   System.out.println (inputLine);
  }
  in.close();
 } catch (Exception e) {
  e.printStackTrace();
 }
}
}
```

Handling HTTP Requests Using java.net.HttpURLConnection :

HttpURLConnection :- A subclass of 'URL Connection' used for HTT
specific features.

ample - Sending a GET Request.

```
import java.io;
Import java.net;

Public class HttpGetExample {
public static void main (String [] args) {
 try {}{

  URL url = new URL ("http://example.com/api/data");
 HttpURLConnection conn = (HttpURLConnection)url.openConnection();
conn.setRequestMethod ("GET");
```

PAGE NO. .....

```
BufferedReader in = new BufferedReader
String responseLine;
while ((responseLine = in.readLine()) != null) {
System.out.println(responseLine);
 }
in.close();
}
catch (IOException e) {
 e.printStackTrace();
 }
}
}
```