

# OOAD

## MULTI THREADING IN JAVA

NAME: SANMAT SANJAYAKUMAR PAYAGOUDAR

SRN: PES1UG20CS385

For optimal CPU utilization, Java's multithreading feature enables the simultaneous execution of two or more software components, each referred to as a thread. Threads are lightweight processes within a larger process.

Threads can be created using two mechanisms: extending the Thread Class or implementing the Runnable Interface. When extending the Thread class, our class cannot extend any other class due to Java's lack of support for multiple inheritance. However, if we implement the Runnable interface, our class can still extend other base classes.

Implementing the Runnable interface provides an object that can be shared among multiple threads. Although the basic functionality of a thread can be achieved by extending the Thread class, the Runnable interface offers advantages such as greater flexibility and the ability to extend other base classes.

The Thread class provides additional inbuilt methods such as `yield()`, `interrupt()`, etc., which are not available in the Runnable interface.

## Code:

```
import java.io.BufferedReader;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.URL;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

public class DownloadManager {

    public static void main(String[] args) {
        String inputFilename = "urls.txt"; // Change this to the input file name
        BufferedReader reader = null;

        try {
            reader = new BufferedReader(new InputStreamReader(
                DownloadManager.class.getResourceAsStream(inputFilename)));

            String urlStr;
            while ((urlStr = reader.readLine()) != null) {
                Thread t = new Thread(new DownloadThread(urlStr));
                t.start();
            }
        } catch (IOException e) {
            e.printStackTrace();
        } finally {
            if (reader != null) {
                try {
                    reader.close();
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        }
    }

    private static class DownloadThread implements Runnable {
        private String urlStr;

        public DownloadThread(String urlStr) {
            this.urlStr = urlStr;
        }

        @Override
        public void run() {
            try {
                URL url = new URL(urlStr);
                String filename = getFileNameFromUrl(url);
                FileOutputStream outputStream = new FileOutputStream(filename);

                byte[] buffer = new byte[4096];
                int bytesRead;
                while ((bytesRead = url.openStream().read(buffer)) != -1) {
                    outputStream.write(buffer, 0, bytesRead);
                }
            }
        }
    }
}
```

```

        outputStream.close();

        System.out.println("Downloaded " + filename + " using Thread " +
Thread.currentThread().getId());
    } catch (IOException e) {
        System.err.println("Error downloading " + urlStr + ": " + e.getMessage());
    }
}

private String getFileNameFromUrl(URL url) {
    Pattern p = Pattern.compile("[^/]*$");
    Matcher m = p.matcher(url.getFile());
    m.find();
    return m.group();
}
}
}

```

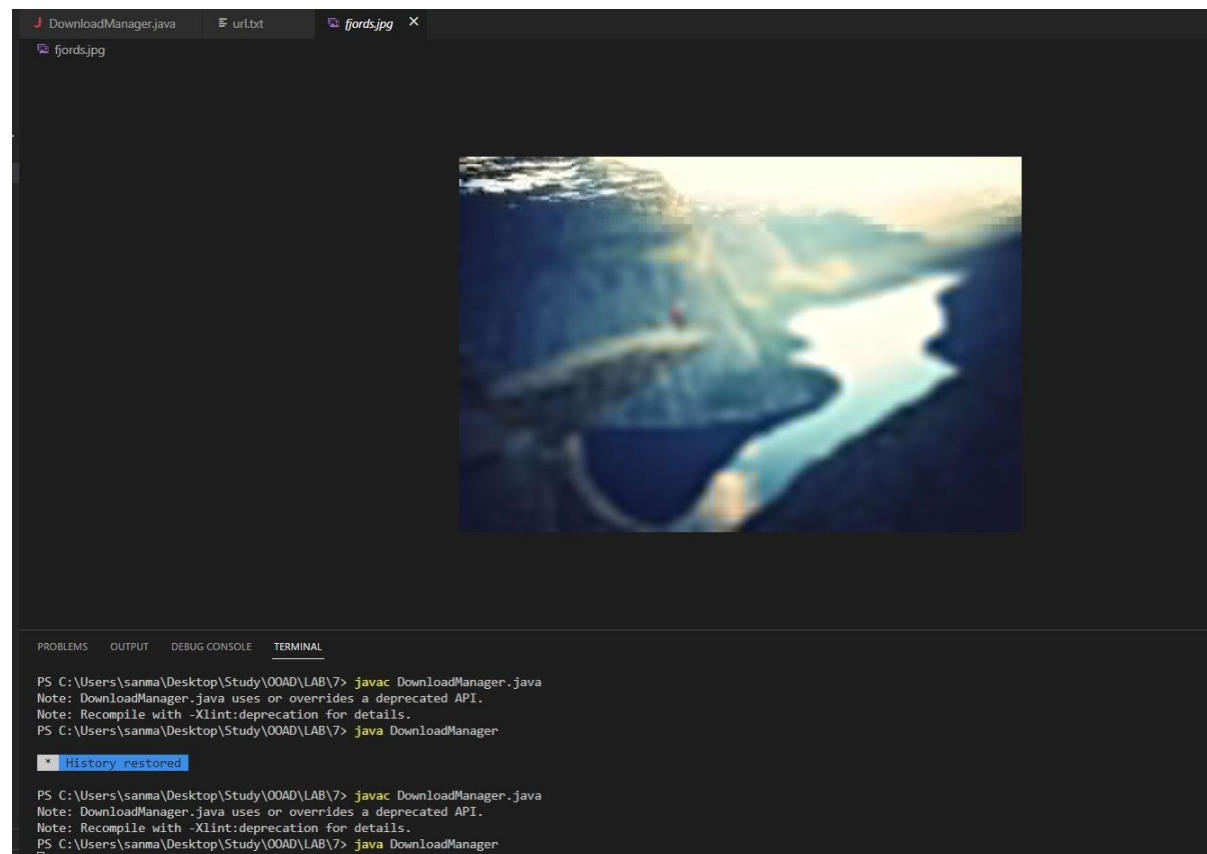
### URLS:

```

url.txt
1 https://example-files.online-convert.com/document/txt/example.txt
2 https://cdn.discordapp.com/attachments/1061299396291080273/1081288165760978965/UE20CS352_Unit3_Slides.pdf
3 https://www.w3schools.com/w3images/fjords.jpg
4 https://file-examples.com/storage/fef1706276640fa2f99a5a4/2017/04/file_example_MP4_480_1_5MG.mp4
5 https://en.wikipedia.org/wiki/Maharana_Pratap

```

### IMAGE:



The screenshot shows an IDE window with the following components:

- File Explorer:** Shows a file named `fjords.jpg` in the project directory.
- Image Preview:** A preview of the `fjords.jpg` image, which depicts a scenic view of a fjord with steep, rocky cliffs and a body of water.
- Terminal:** Displays the following output:
 

```

PS C:\Users\sanma\Desktop\Study\OOAD\LAB\7> javac DownloadManager.java
Note: DownloadManager.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
PS C:\Users\sanma\Desktop\Study\OOAD\LAB\7> java DownloadManager

```

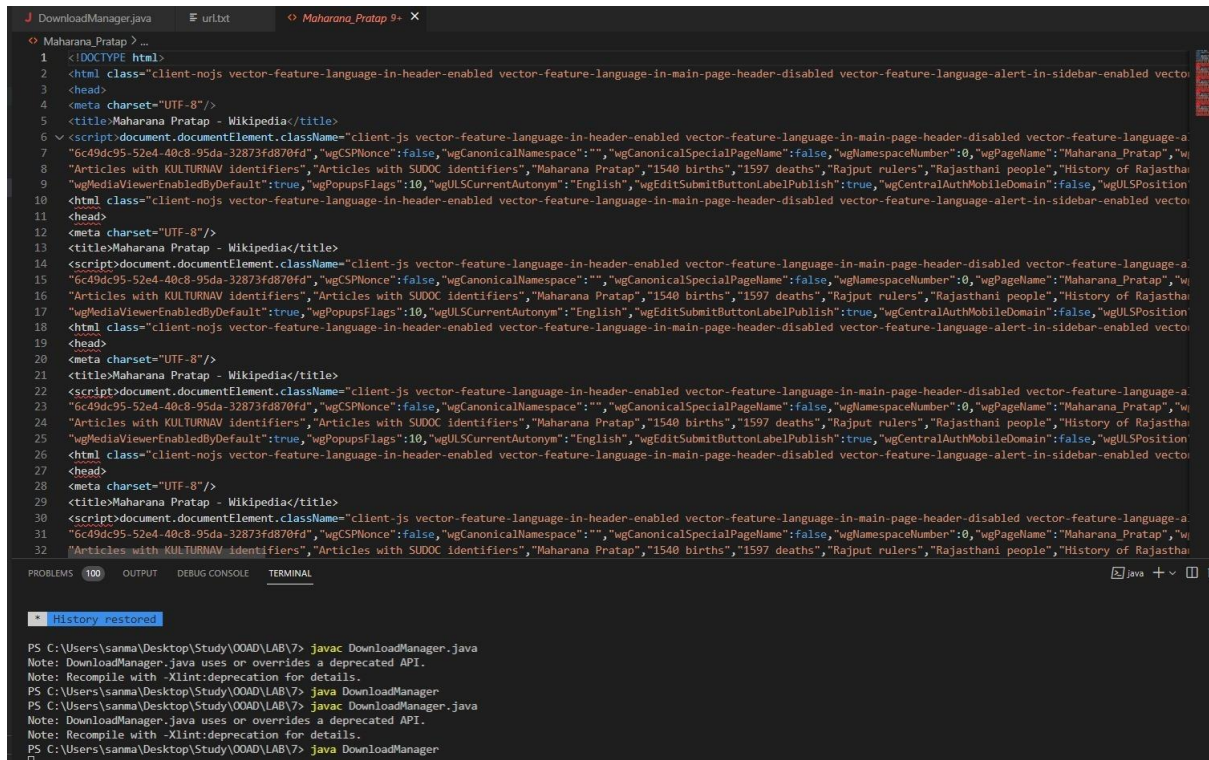
**VIDEO:**



**PDF:**

[illegible]

## HTTP:



The screenshot shows an IDE with a file named `DownloadManager.java` open. The file contains a large block of HTML code, which appears to be a copy-paste of a Wikipedia page for "Maharana Pratap". The code includes a DOCTYPE declaration, a head section with a UTF-8 charset, and a body section with a script tag. The script tag contains a large block of JSON data, which is a copy of the page's metadata and content. The IDE's terminal shows the following output:

```
PS C:\Users\sanna\Desktop\Study\OQAD\LAB\7> javac DownloadManager.java
Note: DownloadManager.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
PS C:\Users\sanna\Desktop\Study\OQAD\LAB\7> java DownloadManager
PS C:\Users\sanna\Desktop\Study\OQAD\LAB\7> javac DownloadManager.java
Note: DownloadManager.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
PS C:\Users\sanna\Desktop\Study\OQAD\LAB\7> java DownloadManager
```

## TEXT FILE:

DownloadManager.java url.txt example.txt X

example.txt

1 TXT test file  
2 Purpose: ProvTXT test file  
3 Purpose: ProvTXT test file  
4 Purpose: ProvTXT test file  
5 Purpose: ProvTXT test file  
6 Purpose: ProvTXT test file  
7 Purpose: ProvTXT test file  
8 Purpose: ProvTXT test file  
9 Purpose: ProvTXT test file  
10 Purpose: ProvTXT test file  
11 Purpose: ProvTXT test file  
12 Purpose: ProvTXT test file  
13 Purpose: ProvTXT test file  
14 Purpose: ProvTXT test file  
15 Purpose: ProvTXT test file  
16 Purpose: ProvTXT test file  
17 Purpose: ProvTXT test file  
18 Purpose: ProvTXT test file  
19 Purpose: ProvTXT test file  
20 Purpose: ProvTXT test file  
21 Purpose: ProvTXT test file  
22 Purpose: ProvTXT test file  
23 Purpose: ProvTXT test file  
24 Purpose: ProvTXT test file  
25 Purpose: ProvTXT test file  
26 Purpose: ProvTXT test file  
27 Purpose: ProvTXT test file  
28 Purpose: ProvTXT test file  
29 Purpose: ProvTXT test file  
30 Purpose: ProvTXT test file  
31 Purpose: ProvTXT test file  
32 Purpose: ProvTXT test file

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS C:\Users\sanma\Desktop\Study\OOAD\LAB\7> javac DownloadManager.java  
Note: DownloadManager.java uses or overrides a deprecated API.  
Note: Recompile with -Xlint:deprecation for details.  
PS C:\Users\sanma\Desktop\Study\OOAD\LAB\7> java DownloadManager

\* History restored

PS C:\Users\sanma\Desktop\Study\OOAD\LAB\7> javac DownloadManager.java  
Note: DownloadManager.java uses or overrides a deprecated API.  
Note: Recompile with -Xlint:deprecation for details.  
PS C:\Users\sanma\Desktop\Study\OOAD\LAB\7> java DownloadManager