Assignment 6:

Sample log file

2024-05-24 14:32:10 INFO Starting the service

2024-05-24 14:33:45 ERROR An error occurred in the service

2024-05-24 14:35:12 INFO Service running smoothly

2024-05-24 14:36:05 ERROR Another error occurred

Use 'grep' to extract lines containing "ERROR".

Use 'awk' to print the date, time, and error message.

Optionally, use 'sed' for further processing if needed.

```
# Check if the log file is provided as an argument
if [ -z "$1" ]; then
  echo "Usage: $0 <logfile>"
  exit 1
fi

LOGFILE="$1"

# Extract lines containing "ERROR" and process with awk
grep "ERROR" "$LOGFILE" | awk '{print $1, $2, $3, $4, $5}'

#Run the script with your log file
./extract_errors.sh sample.log

#Given the example log file provided, the output would be
2024-05-24 14:33:45 ERROR An error occurred in the service
2024-05-24 14:36:05 ERROR Another error occurred
```

#This output shows the date, time, and error message for each line containing "ERROR".

#If additional text processing is needed, sed can be incorporated. For example, to remove extra spaces in the error message:

```
#!/bin/bash

# Check if the log file is provided as an argument
if [ -z "$1" ]; then
  echo "Usage: $0 <logfile>"
  exit 1
fi

LOGFILE="$1"

# Extract lines containing "ERROR", process with awk, and further refine with sed
grep "ERROR" "$LOGFILE" | awk '{print $1, $2, $3, $4, substr($0, index($0,$5))}'
| sed 's/ \+/ /g'
```

```
mohana@LAPTOP-C6DK3IAD:~$ ls
TestDir  class.text  class.txt  mohana  mohana.txt  msen  test.txt
mohana@LAPTOP-C6DK3IAD:~$ mkdir log_processing
mohana@LAPTOP-C6DK3IAD:~$ cd log_processing
mohana@LAPTOP-C6DK3IAD:~/log_processing$ nano sample.log
mohana@LAPTOP-C6DK3IAD:~/log_processing$ grep "ERROR" sample.log > errors.log
mohana@LAPTOP-C6DK3IAD:~/log_processing$ awk '{print $1, $2, $0}' errors.log
```