**3. Creation of Formatter object**

```
formatter = logging.Formatter('%(asctime)s - %(name)s - %(levelname)s: %(message)s',
datefmt='%d/%m/%Y %I:%M:%S %p')
```

**4. Add Formatter to Handler**

```
consoleHandler.setFormatter(formatter)
```

**5. Add Handler to Logger**

```
logger.addHandler(consoleHandler)
```

**6. Write messages by using logger object and the following methods**

```
logger.debug('debug message')
logger.info('info message')
logger.warn('warn message')
logger.error('error message')
logger.critical('critical message')
```

**Note:** Bydefault logger will set to WARNING level. But we can set our own level based on our requirement.

```
logger = logging.getLogger('demologger')
logger.setLevel(logging.INFO)
```

logger log level by default available to console and file handlers. If we are not satisfied with logger level, then we can set log level explicitly at console level and file levels.

```
consoleHandler = logging.StreamHandler()
consoleHandler.setLevel(logging.WARNING)
```

```
fileHandler=logging.FileHandler('abc.log',mode='a')
fileHandler.setLevel(logging.ERROR)
```

**Note:**

console and file log levels should be supported by logger. i.e logger log level should be lower than console and file levels. Otherwise only logger log level will be considered.

**Eg:**

logger==>DEBUG console===>INFO -------->Valid and INFO will be considered
logger==>INFO console===>DEBUG -------->Invalid and only INFO will be considered to the console.