

# Logging

Madhavan Mukund

<https://www.cmi.ac.in/~madhavan>

Programming Concepts using Java

Week 7

# Diagnostic messages

- Typical to generate messages within code for diagnosis

# Diagnostic messages

- Typical to generate messages within code for diagnosis
- Naive approach is to use print statements
  - Need to add / subtract as we go along
  - Enable and disable explicitly

# Diagnostic messages

- Typical to generate messages within code for diagnosis
- Naive approach is to use print statements
  - Need to add / subtract as we go along
  - Enable and disable explicitly
- Instead **log** diagnostic messages separately
  - Logs are arranged hierarchically — choose the level of logging needed
  - Can be displayed in different formats
  - Logs can be processed by other code — **handlers**
    - Can filter out uninteresting entries
  - Logging controlled by a configuration file

- Simplest: call `info()` method of global logger:

```
Logger.getLogger().info("Edit->Copy menu item selected");
```

# Logging

- Simplest: call `info()` method of global logger:

```
Logger.getGlobal().info("Edit->Copy menu item selected");
```

- This prints the following

```
January 10, 2022 10:12:15 PM LoggingImageViewer myFunction  
INFO: Edit->Copy menu item selected
```

# Logging

- Simplest: call `info()` method of global logger:

```
Logger.getGlobal().info("Edit->Copy menu item selected");
```

- This prints the following

```
January 10, 2022 10:12:15 PM LoggingImageViewer myFunction  
INFO: Edit->Copy menu item selected
```

- Suppress logging by executing the following code

```
Logger.getGlobal().setLevel(Level.OFF);
```

# Logging

- Simplest: call `info()` method of global logger:

```
Logger.getGlobal().info("Edit->Copy menu item selected");
```

- This prints the following

```
January 10, 2022 10:12:15 PM LoggingImageViewer myFunction  
INFO: Edit->Copy menu item selected
```

- Suppress logging by executing the following code

```
Logger.getGlobal().setLevel(Level.OFF);
```

- Create a custom logger

```
private static final Logger myLogger =  
    Logger.getLogger("in.ac.iitm.onlinedegree");
```



# Logging

- Simplest: call `info()` method of global logger:

```
Logger.getGlobal().info("Edit->Copy menu item selected");
```

- This prints the following

```
January 10, 2022 10:12:15 PM LoggingImageViewer myFunction  
INFO: Edit->Copy menu item selected
```

- Suppress logging by executing the following code

```
Logger.getGlobal().setLevel(Level.OFF);
```

- Create a custom logger

```
private static final Logger myLogger =  
    Logger.getLogger("in.ac.iitm.onlinedegree");
```

- Logger names are hierarchical, like package names
- Setting a property for `in.ac.iitm` automatically sets it for `in.ac.iitm.onlinedegree`

# Logging levels

- Seven logging levels
  - SEVERE, WARNING, INFO, CONFIG, FINE, FINER, FINEST

# Logging levels

- Seven logging levels
  - SEVERE, WARNING, INFO, CONFIG, FINE, FINER, FINEST
- By default, first three levels are logged

# Logging levels

- Seven logging levels
  - SEVERE, WARNING, INFO, CONFIG, FINE, FINER, FINEST
- By default, first three levels are logged
- Can set a different level

```
logger.setLevel(Level.FINE);
```

# Logging levels

- Seven logging levels
  - SEVERE, WARNING, INFO, CONFIG, FINE, FINER, FINEST
- By default, first three levels are logged
- Can set a different level

```
logger.setLevel(Level.FINE);
```
- Turn on all levels, or turn off all logging

```
logger.setLevel(Level.ALL);  
logger.setLevel(Level.OFF);
```

# Logging levels

- Seven logging levels
  - SEVERE, WARNING, INFO, CONFIG, FINE, FINER, FINEST
- By default, first three levels are logged
- Can set a different level

```
logger.setLevel(Level.FINE);
```
- Turn on all levels, or turn off all logging

```
logger.setLevel(Level.ALL);  
logger.setLevel(Level.OFF);
```
- Can also change logging properties through a configuration file
  - Look up the documentation

# Summary

- Logging gives us more flexibility and control over tracking diagnostic messages than simple print statements
- Can define a hierarchy of loggers
- Seven levels of messages — control which levels are printed
- Control logging from within code or through external configuration file