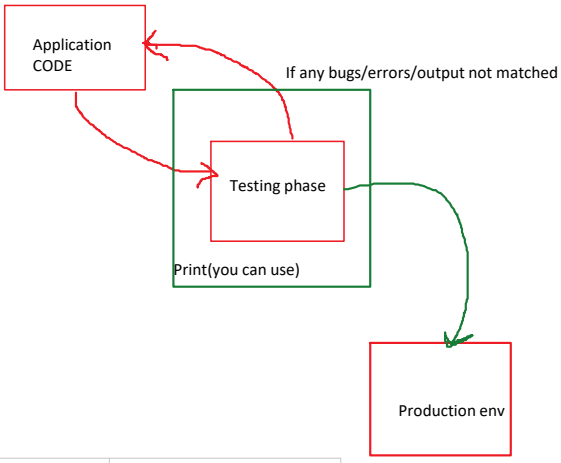
Logging and Debugging

21 January 2026 09:02



Expected/op

Student name	GCP
Ravi	9
Venkat	8
NIRAML	6
Kumar	6
Vinay	5
Murali	10

Actual outp

Student name	GCP
Ravi	9
Venkat	8
NIRAML	6
Kumar	-
Vinay	5
Murali	-

Close
Hi @bgawat ,
When I am texting gcp module I found the bug
/screen short
There is no GCP for following student , kumar,murali
High --
Low
medium

Print()	Logging
Temporary debugging	Permanent tracking is available
No severity levels	Has levels (debug,info..)
Not suitable for production	In real time application
We can not write the data in to files easily	Easy to write to files

I will be checking on it and get back to you ..

Modules are modified please start testing on it

What is Logging :
Logging is the process of recording important information while program runs
These records helps to developers and devops eng..

- **Diagnose issues**
- **Trace failure**
- **Monitor system behaviors**
- **Store execution history**
- **Investigate security failures**
- **Perform post-mortem analysis**

Logging is critical in production , especially for :
Micro services
CI/Cd pipelines
Automation scripting
Cloud deployments
Container-based systems

Why logging is needed?

- **When the software is runs In production**
- **Theree is no developer watching the screen**
- **Error automatically captured'**
- **Logs are used to debug, auditing, monitoring**
- **Logs are act as cctv footage of programs**

Import logging (module)

Log levels
Log formatting
Log files
Rotate logs
Multiple handlers

```
import logging
# logging.basicConfig(level=logging.DEBUG)
# logging.debug("this is a debug message ")
# logging.info(" Application started ")
# logging.warning("low memory waring")
# logging.error("failed to procvess file")
# logging.critical("system crashes ")
#logging in to files :
logging.basicConfig(filename="app.log",level=logging.WARNING,format="%(asctime)
s - %(levelname)s - %(message)s")
logging.debug("this is a debug message ")
logging.info(" Application started ")
logging.warning("low memory waring")
logging.error("failed to procvess file")
logging.critical("system crashes ")
```

Priority:
By default **WARNING** level

VALUE	LEVEL	LOGGING FUNCTION	DESCRIPTION
10	**DEBUG**	**Logging.debug()**	**Lowest level and used to record simple details**
20	**INFO**	**Logging.info()**	**Record general information**
30	**WARNING**	**Logging.warning()**	**Potential issues which may not cause error in the future**
40	**ERROR**	**Logging.error()**	**Record error which cause a section of the code fail**
50	**CRITICAL**	**Logging.crititcal()**	**Highest Level , blockers which fails your whole program**

Debugging

1. Print(")
2. Logging.debug({variable},
3. Vs code break point Using Breakpoints
4. Using Python Debugger (pdb)