Software Engineering

UE23CS341A

5th Semester, Academic Year 2023

Date: 27/10/2025

Name: Roshini Ramesh	SRN: PES1UG23CS488	Section: H

LAB 4: STATIC CODE ANALYSIS

1. **Initial behaviour when the program is first run:**

```
Traceback (most recent call last):
  File "/workspaces/inventory-system/inventory_system.py", line 61, in <module>
    main()
  File "/workspaces/inventory-system/inventory_system.py", line 51, in main
    addItem(123, "ten")  # invalid types, no check
    ^^^^^^^^^^^^^^^^^^^
  File "/workspaces/inventory-system/inventory_system.py", line 11, in addItem
    stock_data[item] = stock_data.get(item, 0) + qty
                       ~~~~~~~~~~~~~~~~~~~~~~~~~^~~~~
TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

2. **Reports generated by the static code analysis tools:**

- bandit_report.txt

```
Run started:2025-10-27 03:19:57.071646

Test results:
>> Issue: [B110:try_except_pass] Try, Except, Pass detected.
   Severity: Low   Confidence: High
   CWE: CWE-703 (https://cwe.mitre.org/data/definitions/703.html)
   More Info: https://bandit.readthedocs.io/en/1.8.6/plugins/b110_try_except_pass.html
   Location: ./inventory_system.py:19:4
18                  del stock_data[item]
19              except:
```

```
20              pass
21

--------------------------------------------------
>> Issue: [B307:blacklist] Use of possibly insecure function - consider using safer ast.literal_eval.
   Severity: Medium   Confidence: High
   CWE: CWE-78 (https://cwe.mitre.org/data/definitions/78.html)
   More Info: https://bandit.readthedocs.io/en/1.8.6/blacklists/blacklist_calls.html#b307-eval
   Location: ./inventory_system.py:59:4
58          printData()
59          eval("print('eval used')")  # dangerous
60

--------------------------------------------------

Code scanned:
        Total lines of code: 50
        Total lines skipped (#nosec): 0
        Total potential issues skipped due to specifically being disabled (e.g., #nosec BXXX): 0

Run metrics:
        Total issues (by severity):
                Undefined: 0
                Low: 1
                Medium: 1
                High: 0
        Total issues (by confidence):
                Undefined: 0
                Low: 0
                Medium: 0
                High: 2
Files skipped (0):
```

- flake8_report.txt

```
inventory_system.py:2:1: F401 'logging' imported but unused
inventory_system.py:8:1: E302 expected 2 blank lines, found 1
inventory_system.py:14:1: E302 expected 2 blank lines, found 1
inventory_system.py:19:5: E722 do not use bare 'except'
inventory_system.py:22:1: E302 expected 2 blank lines, found 1
inventory_system.py:25:1: E302 expected 2 blank lines, found 1
inventory_system.py:31:1: E302 expected 2 blank lines, found 1
inventory_system.py:36:1: E302 expected 2 blank lines, found 1
inventory_system.py:41:1: E302 expected 2 blank lines, found 1
inventory_system.py:48:1: E302 expected 2 blank lines, found 1
```

inventory_system.py:61:1: E305 expected 2 blank lines after class or function definition, found 1

- pylint_report.txt

********** Module inventory_system
inventory_system.py:1:0: C0114: Missing module docstring (missing-module-docstring)
inventory_system.py:8:0: C0116: Missing function or method docstring (missing-function-docstring)
inventory_system.py:8:0: C0103: Function name "addItem" doesn't conform to snake_case naming style (invalid-name)
inventory_system.py:8:0: W0102: Dangerous default value [] as argument (dangerous-default-value)
inventory_system.py:12:16: C0209: Formatting a regular string which could be an f-string (consider-using-f-string)
inventory_system.py:14:0: C0116: Missing function or method docstring (missing-function-docstring)
inventory_system.py:14:0: C0103: Function name "removeItem" doesn't conform to snake_case naming style (invalid-name)
inventory_system.py:19:4: W0702: No exception type(s) specified (bare-except)
inventory_system.py:22:0: C0116: Missing function or method docstring (missing-function-docstring)
inventory_system.py:22:0: C0103: Function name "getQty" doesn't conform to snake_case naming style (invalid-name)
inventory_system.py:25:0: C0116: Missing function or method docstring (missing-function-docstring)
inventory_system.py:25:0: C0103: Function name "loadData" doesn't conform to snake_case naming style (invalid-name)
inventory_system.py:26:8: W1514: Using open without explicitly specifying an encoding (unspecified-encoding)
inventory_system.py:27:4: W0603: Using the global statement (global-statement)
inventory_system.py:26:8: R1732: Consider using 'with' for resource-allocating operations (consider-using-with)
inventory_system.py:31:0: C0116: Missing function or method docstring (missing-function-docstring)
inventory_system.py:31:0: C0103: Function name "saveData" doesn't conform to snake_case naming style (invalid-name)
inventory_system.py:32:8: W1514: Using open without explicitly specifying an encoding (unspecified-encoding)
inventory_system.py:32:8: R1732: Consider using 'with' for resource-allocating operations (consider-using-with)
inventory_system.py:36:0: C0116: Missing function or method docstring (missing-function-docstring)
inventory_system.py:36:0: C0103: Function name "printData" doesn't conform to snake_case naming style (invalid-name)
inventory_system.py:41:0: C0116: Missing function or method docstring (missing-function-docstring)
inventory_system.py:41:0: C0103: Function name "checkLowItems" doesn't conform to snake_case naming style (invalid-name)
inventory_system.py:48:0: C0116: Missing function or method docstring (missing-function-docstring)
inventory_system.py:59:4: W0123: Use of eval (eval-used)
inventory_system.py:2:0: W0611: Unused import logging (unused-import)

Your code has been rated at 4.80/10 (previous run: 4.80/10, +0.00)

3. Known Issue Table:

Issue	Type	Line(s)	Description	Fix Approach
Mutable default argument	Behavioral Bug	8	Dangerous default list shared across calls	Change signature to logs=None and inside: if logs is None: logs = []. Unit test usage.
No type validation	Runtime Bug	addItem ~11; triggered by call at main ~51	addItem(123, "ten") produces TypeError: unsupported operand type(s) for +: 'int' and 'str' because qty is str and item is not string.	Validate inputs: if not isinstance(item, str): raise TypeError(...) and qty = int(qty) or isinstance(qty, int). Add tests and sanitize user input.
Bare except	Bug/ Maintainability	19	except: in removeItem hides all exceptions (including programming errors).	Catch specific exceptions: except KeyError: or except (KeyError, TypeError) as e:. Log or re-raise as appropriate; do not pass silently.
Use of eval()	Security (high)	59	eval("print('eval used')") — allows arbitrary code execution.	Remove eval. If you must parse literals use ast.literal_eval. Prefer explicit code paths.
Unused import logging	Style/ Cleanliness	2	logging imported but never used.	Remove unused import or add appropriate logging usage (e.g., logging.getLogger(__name__)) and log exceptions/info.
Missing blank lines	Style	8, 14, 22, 25, 31, 36, 41, 48	Functions and definitions not separated by required blank lines	Add blank lines between top-level function definitions (2 blank lines).
Missing blank lines between top-level function definitions	Style	61	Reported by flake8 at file end — ensure blank lines following definitions.	Add required blank lines
Missing module docstring	Style/ Maintainability	module start	No module-level docstring describing purpose.	Add module docstring at top: brief description, author, usage.

Missing function docstrings	Style/ Maintainability	many functions (8, 14, 22, 25, 31, 36, 41, 48)	Functions lack docstrings describing args/returns/raises.	Add concise docstrings for each function describing params, return, exceptions.
Non-snake_case function names	Style/ Lint	functions at ~8 (addItem), ~14 (removeItem), ~22 (getQty), ~25 (loadData), ~31 (saveData), ~36 (printData), ~41 (checkLowItems)	Function names use camelCase / mixedCase instead of snake_case	ename to snake_case (add_item, remove_item, get_qty, load_data, save_data, print_data, check_low_items) and adjust all call sites.
Formatting using % instead of f-string	Style/ Suggestion	12	"%s: Added %d of %s" % (...) recommended to use f-strings for readability.	Replace with f"{datetime.now()}: Added {qty} of {item}".
open() without encoding	Reliability/ Portability	26 (loadData) and 32 (saveData)	open(file, "r") and open(file, "w") with no explicit encoding can cause platform differences.	Use with open(file, "r", encoding="utf-8") as f: and with open(file, "w", encoding="utf-8") as f:.
Not using with context manager for file operations	Resource handling	26–33	Manual open/close — if exception occurs, files may remain open.	Use with open(...) as f: to ensure automatic close; handle FileNotFoundError and json.JSONDecodeError.
Use of global stock_data	Design/ Maintainability	26–28	loadData uses global and reassigns stock_data, which can surprise callers and complicate testing.	Avoid global. Return loaded dict, or better: stock_data.clear(); stock_data.update(loaded). Encapsulate state in a class.
getQty raises KeyError on missing item	Runtime Bug	22	return stock_data[item] will raise KeyError if item not present. Pylint flagged missing docstring but not this exact bug.	Use return stock_data.get(item, 0) or explicitly raise KeyError("...") with clear message. Document behavior.
JSON load/save without error handling	Reliability	26–33	json.loads and json.dumps not protected against JSONDecodeError or I/O errors; save	Wrap in try/except to handle json.JSONDecodeError, IOError. Use atomic write (write to temp file and os.replace).

			can corrupt file partially.	
removeItem negative/over-removal logic	Logic / Bug	14-19	stock_data[item] -= qty may allow negative counts before deletion; silent except hides errors.	Validate qty >= 0, check existence of item before decreasing, return status or raise ValueError on invalid remove.
Iteration over dict without deterministic order	UX/Minor	41	checkLowItems returns list of keys only; no quantities or deterministic sort.	Return [(item, qty), ...] and optionally sort (e.g., sorted(..., key=lambda t: t[0])) for deterministic reports.
main() runs on import (no if __name__ == "__main__":)	Maintainability / Safety	48-61	Module executes demo code and side-effects on import (saves/loads files, uses eval). Hard to import for tests.	Protect with if __name__ == "__main__": main() and remove unsafe demo calls (eval, forced save/load).
eval in code flagged by Bandit & Pylint (duplicate)	Security	59	(duplicate of earlier but important)	Remove eval. Use ast.literal_eval for literal parsing or explicit code.