**Source Code**

import os

import shutil

import unittest

class DiskSpaceManager:

def \_init\_(self, path):

self.path = path

self.total, self.used, self.free = shutil.disk\_usage(path)

def get\_disk\_usage(self):

return {

'total': self.total,

'used': self.used,

'free': self.free

}

def analyze\_disk\_space(self):

usage\_percentage = (self.used / self.total) \* 100

return {

'total\_space': self.total,

'used\_space': self.used,

'free\_space': self.free,

'usage\_percentage': usage\_percentage

}

def suggest\_cleanup\_actions(self):

suggestions = []

usage\_info = self.analyze\_disk\_space()

if usage\_info['usage\_percentage'] > 80:

suggestions.append("Consider removing unnecessary files.")

if usage\_info['free\_space'] < 10 \* (1024 \*\* 3): # If less than 10 GB is free

suggestions.append("Free up some space to avoid running out of disk.")

if not suggestions:

suggestions.append("Disk space usage is optimal.")

return suggestions

class TestDiskSpaceManager(unittest.TestCase):

def setUp(self):

self.manager = DiskSpaceManager("/")

def test\_disk\_usage(self):

usage = self.manager.get\_disk\_usage()

self.assertIn('total', usage)

self.assertIn('used', usage)

self.assertIn('free', usage)

def test\_cleanup\_suggestions(self):

suggestions = self.manager.suggest\_cleanup\_actions()

self.assertTrue(len(suggestions) > 0)

if \_\_name\_\_ == "\_main\_":

unittest.main()

if \_\_name\_\_ == "\_main\_":

manager = DiskSpaceManager("/")

disk\_usage = manager.get\_disk\_usage()

print("Disk Usage Details:")

print(disk\_usage)

analysis = manager.analyze\_disk\_space()

print("\nDisk Space Analysis:")

print(f"Total Space: {analysis['total\_space']} bytes")

print(f"Used Space: {analysis['used\_space']} bytes")

print(f"Free Space: {analysis['free\_space']} bytes")

print(f"Usage Percentage: {analysis['usage\_percentage']:.2f}%")