class ActivePowerFilter:

def \_init\_(self, rated\_voltage, rated\_current, frequency):

self.rated\_voltage = rated\_voltage

self.rated\_current = rated\_current

self.frequency = frequency

def reduce\_harmonics(self, input\_voltage, input\_current):

# Implement active filtering algorithm to reduce harmonics

# This can involve techniques like p-q theory, hysteresis control, etc.

# For simplicity, let's assume it reduces harmonics by a fixed percentage

harmonic\_reduction\_percentage = 0.2 # 20% reduction

output\_voltage = input\_voltage \* (1 - harmonic\_reduction\_percentage)

output\_current = input\_current \* (1 - harmonic\_reduction\_percentage)

return output\_voltage, output\_current

# Example usage:

def main():

rated\_voltage = 220 # Rated voltage of the system (in volts)

rated\_current = 100 # Rated current of the system (in amps)

frequency = 50 # Frequency of the system (in Hz)

active\_filter = ActivePowerFilter(rated\_voltage, rated\_current, frequency)

input\_voltage = 230 # Input voltage with harmonics (in volts)

input\_current = 110 # Input current with harmonics (in amps)

output\_voltage, output\_current = active\_filter.reduce\_harmonics(input\_voltage, input\_current)

print("Original voltage:", input\_voltage)

print("Filtered voltage:", output\_voltage)

print("Original current:", input\_current)

print("Filtered current:", output\_current)

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

    main()