# Rule 1: Only lowercase letters

if password.islower():

score = 6

# Rule 2: Has lowercase, uppercase, and numbers

if any(c.islower() for c in password) and \

any(c.isupper() for c in password) and \

any(c.isdigit() for c in password):

score = 8

# Rule 3: Has lowercase, uppercase, numbers, and special characters

if any(c.islower() for c in password) and \

any(c.isupper() for c in password) and \

any(c.isdigit() for c in password) and \

any(not c.isalnum() for c in password): # special characters

score = 10

# Check the score and print result

if score == 6:

print("Password is week")

elif score == 8:

print("Password is strong")

elif score == 10:

print("Password is very strong")

# Test the function

check\_password\_strength("tiger") # Output: week

check\_password\_strength("LioN123") # Output: strong

check\_password\_strength("LionTig@r123") # Output: very strong

Inheritance:

class Grandfather:

def grandfather\_method(self):

return "This is grandfather method"

class Father(Grandfather): # Inherits from Grandfather

def father\_method(self):

return "This is father method"

class Mother:

def mother\_method(self):

return "This is mother method"

class Child(Father, Mother): # Inherits from both Father and Mother

def child\_method(self):

return "This is child method"

# Creating proper object instances

grandfather\_object = Grandfather()

father\_object = Father()

mother\_object = Mother()

child\_object = Child()

# Accessing methods correctly

print(grandfather\_object.grandfather\_method())

print(father\_object.father\_method())

print(father\_object.grandfather\_method()) # Inherited

print(mother\_object.mother\_method())

print(child\_object.child\_method())

print(child\_object.father\_method()) # Inherited

print(child\_object.grandfather\_method()) # Inherited

print(child\_object.mother\_method())