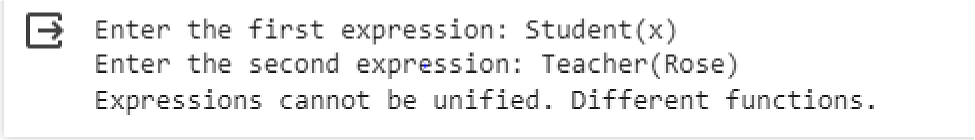
WEEK 8

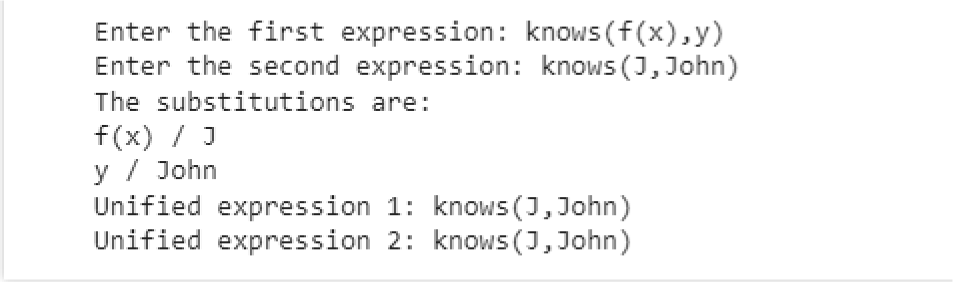
Implement unification in first order logic.

```
def unify(expr1, expr2):
    # Split expressions into function and arguments
    func1, args1 = expr1.split('(', 1)
    func2, args2 = expr2.split('(', 1)

    # Check if functions are the same
    if func1 != func2:
        print("Expressions cannot be unified. Different functions.")
        return None

    args1 = args1.rstrip(')').split(',')
    args2 = args2.rstrip(')').split(',')

    substitution = {}

    # Unify arguments
    for a1, a2 in zip(args1, args2):
        if a1.islower() and a2.islower() and a1 != a2:
            substitution[a1] = a2
        elif a1.islower() and not a2.islower():
            substitution[a1] = a2
        elif not a1.islower() and a2.islower():
            substitution[a2] = a1
        elif a1 != a2:
            print("Expressions cannot be unified. Incompatible arguments.")
            return None

    return substitution


def apply_substitution(expr, substitution):
    for key, value in substitution.items():
        expr = expr.replace(key, value)
    return expr


# Main program
if __name__ == "__main__":
    # Sample input
    expr1 = input("Enter the first expression: ")
    expr2 = input("Enter the second expression: ")

    # Unify expressions
    substitution = unify(expr1, expr2)

    # Display result
    if substitution:
        print("The substitutions are:")
        for key, value in substitution.items():
            print(f'{key} / {value}')

        # Apply substitution to original expressions
        expr1_result = apply_substitution(expr1, substitution)
        expr2_result = apply_substitution(expr2, substitution)

        print(f'Unified expression 1: {expr1_result}')
        print(f'Unified expression 2: {expr2_result}')
```

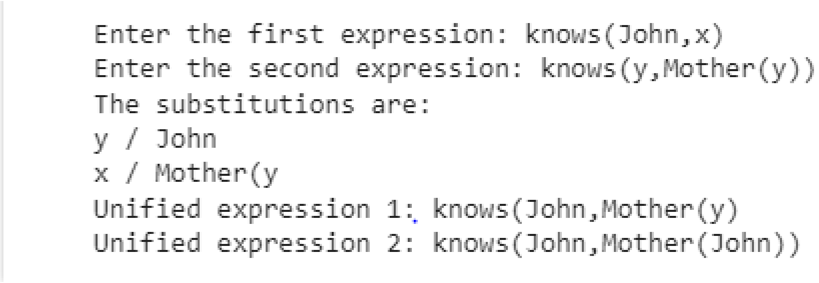
Output:



```
Enter the first expression: Student(x)
Enter the second expression: Teacher(Rose)
Expressions cannot be unified. Different functions.
```



```
Enter the first expression: knows(f(x),y)
Enter the second expression: knows(J,John)
The substitutions are:
f(x) / J
y / John
Unified expression 1: knows(J,John)
Unified expression 2: knows(J,John)
```



```
Enter the first expression: knows(John,x)
Enter the second expression: knows(y,Mother(y))
The substitutions are:
y / John
x / Mother(y
Unified expression 1: knows(John,Mother(y)
Unified expression 2: knows(John,Mother(John))
```