Assignment one

Name: **Sewbesew Yimer**

Question: Assuming a list of relations is given as a list, write a Python and PL SQL class that generates HORIZONTAL miniterm fragments from a list of predicates Pr.

Solution:

Python Implementation

```
class MinitermGenerator:
    def __init__(self, predicates):
        self.predicates = predicates
    def generate_miniterms(self):
        miniterms = []
        num_predicates = len(self.predicates)
        # Generate miniterm fragments
        for i in range(1 << num_predicates):  # 2^num_predicates combinations
            fragment = []
            for j in range(num_predicates):
                # Check if the j-th predicate is included in this miniterm
                if i & (1 << j):
                    fragment.append(self.predicates[j])
            miniterms.append(fragment)
        return miniterms
```

PL/SQL Implementation

```
CREATE OR REPLACE TYPE miniterm_fragment AS TABLE OF VARCHAR2(100);
/
CREATE OR REPLACE PROCEDURE generate_miniterms(predicates IN miniterm_fragment,
miniterms OUT miniterm_fragment) IS
    num_predicates INTEGER := predicates.COUNT;
    combination_count INTEGER := POWER(2, num_predicates);
BEGIN
    miniterms := miniterm_fragment();
    FOR i IN 0 .. combination_count - 1 LOOP
    DECLARE
      fragment VARCHAR2(4000) := ";
    BEGIN
      FOR j IN 0 .. num_predicates - 1 LOOP
        IF BITAND(i, POWER(2, j)) != 0 THEN
          fragment := fragment || predicates(j + 1) || ' ';
        END IF;
      END LOOP;
      -- Add the fragment to the output if it's not empty
      IF fragment IS NOT NULL THEN
        miniterms.EXTEND;
        miniterms(miniterms.COUNT) := TRIM(fragment);
      END IF;
    END;
  END LOOP;
END;
/
```