Pandit Deendayal Energy University, Gandhinagar

School of Technology

Department of Computer Science & Engineering

System Software & Compiler Design Lab (20CP302P)

Name: **Sutariya Smit Dharmendrabhai**

Enrolment No: **21BCP142**

Semester: **V**

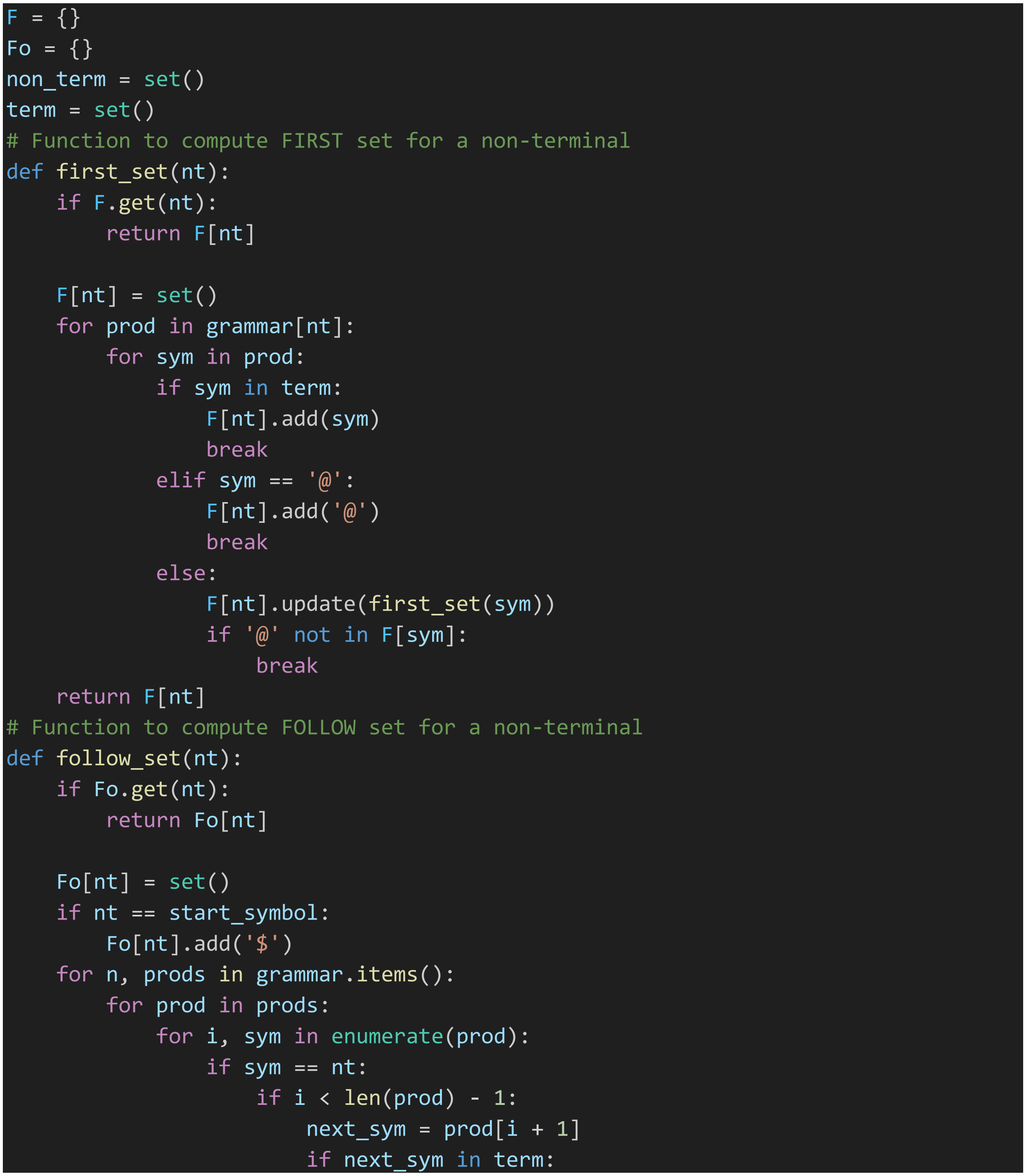
Division: **3 (G5)**

Branch: **Computer Science Engineering**

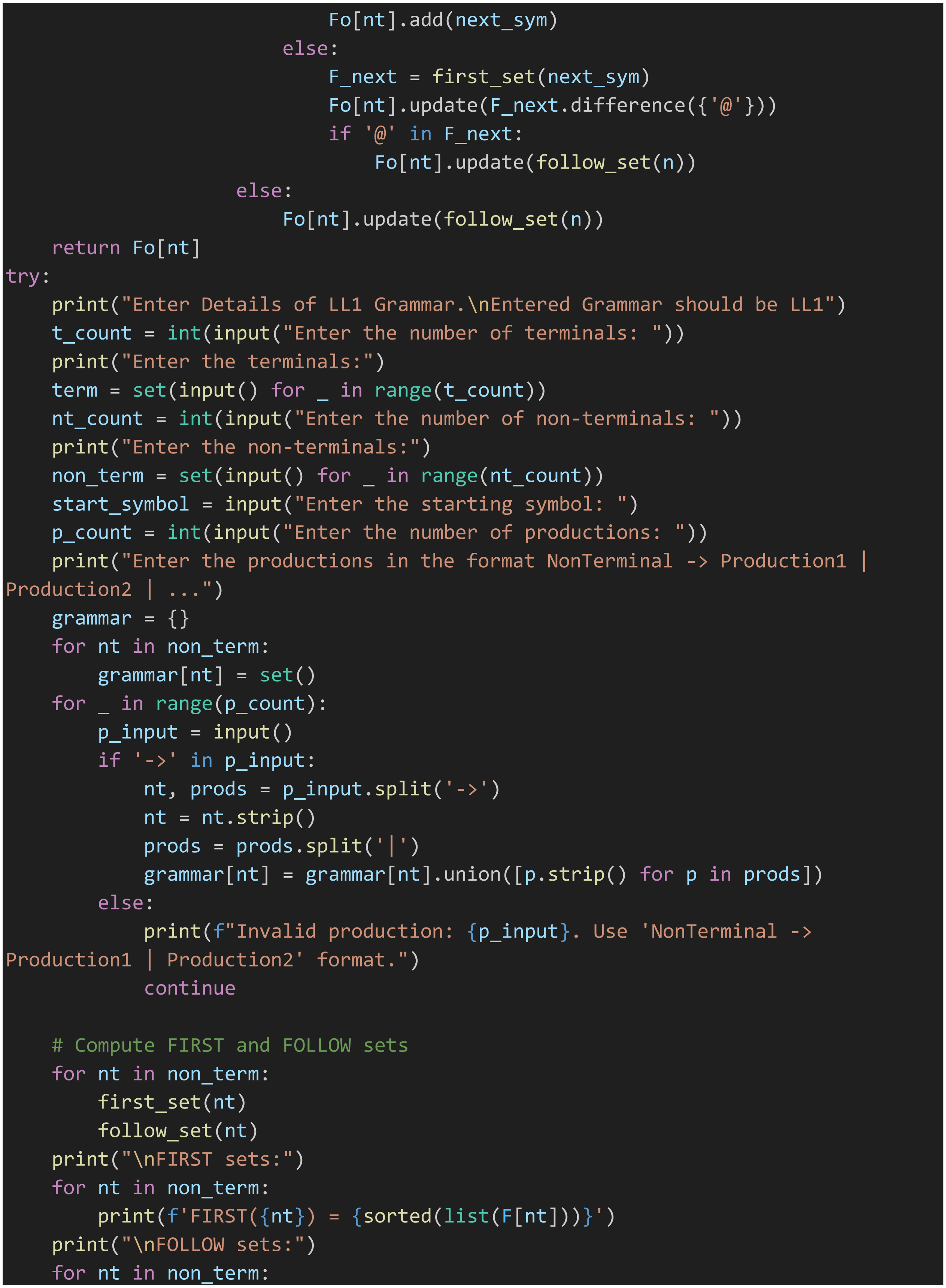
Practical: 5

Aim: Write a program to calculate first and follow of a given LL (1) grammar.

Code:

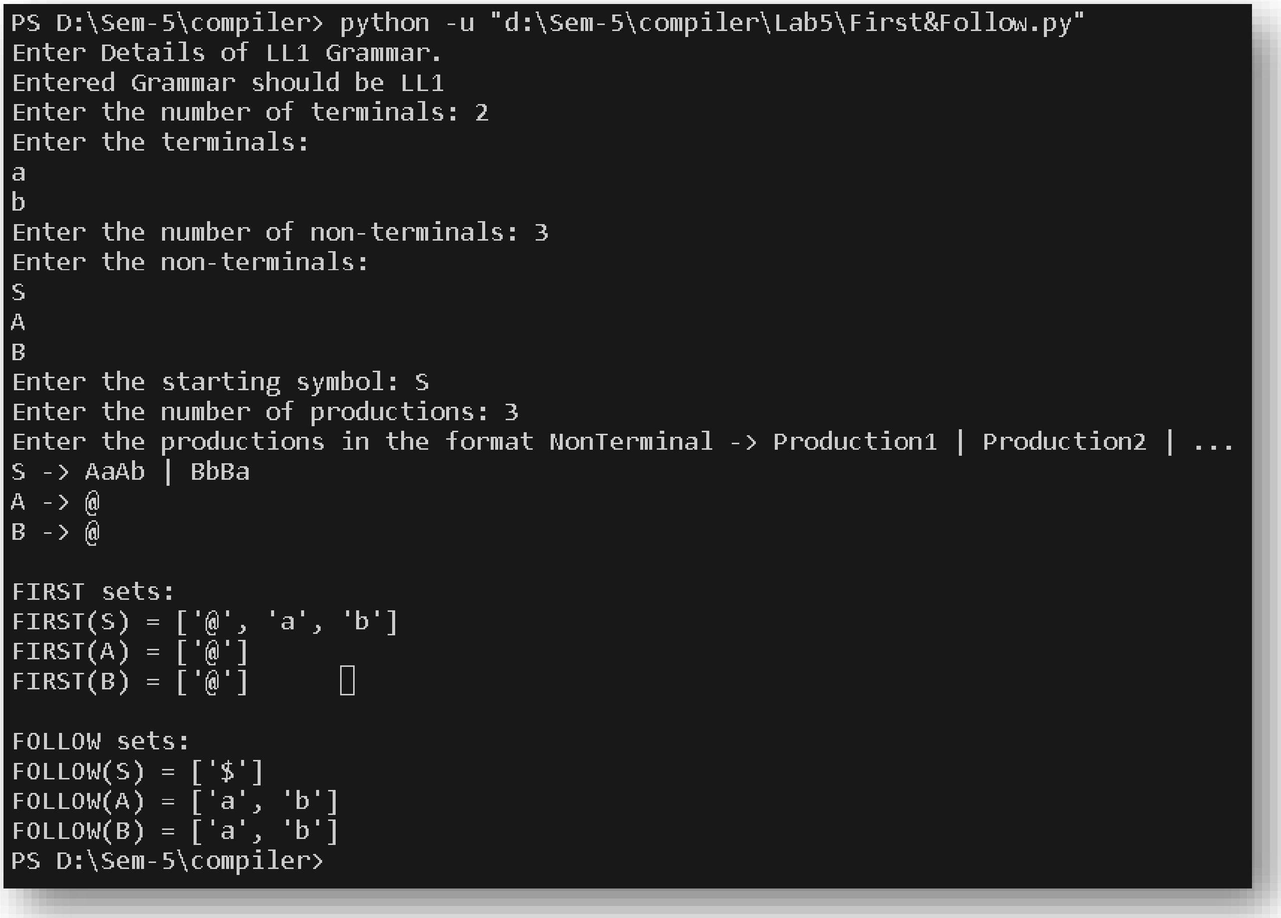


```
F = {}
Fo = {}
non_term = set()
term = set()
# Function to compute FIRST set for a non-terminal
def first_set(nt):
    if F.get(nt):
        return F[nt]

    F[nt] = set()
    for prod in grammar[nt]:
        for sym in prod:
            if sym in term:
                F[nt].add(sym)
                break
            elif sym == '@':
                F[nt].add('@')
                break
            else:
                F[nt].update(first_set(sym))
                if '@' not in F[sym]:
                    break
    return F[nt]
# Function to compute FOLLOW set for a non-terminal
def follow_set(nt):
    if Fo.get(nt):
        return Fo[nt]

    Fo[nt] = set()
    if nt == start_symbol:
        Fo[nt].add('$')
    for n, prods in grammar.items():
        for prod in prods:
            for i, sym in enumerate(prod):
                if sym == nt:
                    if i < len(prod) - 1:
                        next_sym = prod[i + 1]
                        if next_sym in term:
```



```
                            Fo[nt].add(next_sym)
                        else:
                            F_next = first_set(next_sym)
                            Fo[nt].update(F_next.difference({'@'}))
                            if '@' in F_next:
                                Fo[nt].update(follow_set(n))
                    else:
                        Fo[nt].update(follow_set(n))
    return Fo[nt]
try:
    print("Enter Details of LL1 Grammar.\nEntered Grammar should be LL1")
    t_count = int(input("Enter the number of terminals: "))
    print("Enter the terminals:")
    term = set(input() for _ in range(t_count))
    nt_count = int(input("Enter the number of non-terminals: "))
    print("Enter the non-terminals:")
    non_term = set(input() for _ in range(nt_count))
    start_symbol = input("Enter the starting symbol: ")
    p_count = int(input("Enter the number of productions: "))
    print("Enter the productions in the format NonTerminal -> Production1 |
Production2 | ...")
    grammar = {}
    for nt in non_term:
        grammar[nt] = set()
    for _ in range(p_count):
        p_input = input()
        if '->' in p_input:
            nt, prods = p_input.split('->')
            nt = nt.strip()
            prods = prods.split('|')
            grammar[nt] = grammar[nt].union([p.strip() for p in prods])
        else:
            print(f"Invalid production: {p_input}. Use 'NonTerminal ->
Production1 | Production2' format.")
            continue

    # Compute FIRST and FOLLOW sets
    for nt in non_term:
        first_set(nt)
        follow_set(nt)
    print("\nFIRST sets:")
    for nt in non_term:
        print(f'FIRST({nt}) = {sorted(list(F[nt]))}')
    print("\nFOLLOW sets:")
    for nt in non_term:
        print(f'FOLLOW({nt}) = {sorted(list(Fo[nt]))}')
except Exception as e:
    print(f"An error occurred: {e}")
```

Output:



```
PS D:\Sem-5\compiler> python -u "d:\Sem-5\compiler\Lab5\First&Follow.py"
Enter Details of LL1 Grammar.
Entered Grammar should be LL1
Enter the number of terminals: 2
Enter the terminals:
a
b
Enter the number of non-terminals: 3
Enter the non-terminals:
S
A
B
Enter the starting symbol: S
Enter the number of productions: 3
Enter the productions in the format NonTerminal -> Production1 | Production2 | ...
S -> AaAb | BbBa
A -> @
B -> @

FIRST sets:
FIRST(S) = ['@', 'a', 'b']
FIRST(A) = ['@']
FIRST(B) = ['@']

FOLLOW sets:
FOLLOW(S) = ['$']
FOLLOW(A) = ['a', 'b']
FOLLOW(B) = ['a', 'b']
PS D:\Sem-5\compiler>
```