Compiler Construction

LAB MID

By

M Bilal Younas

FA19-BCS-062

Section - B

Submitted to,

Dr. Bilal Haider

Department of Computer Science

COMSATS University Islamabad,

Attock Campus

Q2

CODE :-

```
class Parser:
    def __init__(self, input_string):
        self.tokens = list(input_string)
        self.current_token = None

    def match(self, expected_token):
        if self.current_token == expected_token:
            self.consume()
        else:
            raise ValueError(f"Expected {expected_token}, got {self.current_token}")

    def consume(self):
        if self.tokens:
            self.current_token = self.tokens.pop(0)
        else:
            self.current_token = None

    def parse(self):
        self.consume()  # Initialize current_token
        self.parse_S()
        print("Parsing successful!")

    def parse_S(self):
        self.parse_E()
        self.match('$')

    def parse_E(self):
        self.parse_T()
        self.parse_E_prime()

    def parse_E_prime(self):
        if self.current_token == '+':
            self.consume()
            self.parse_T()
            self.parse_E_prime()
        # else, it's ε (epsilon), do nothing

    def parse_T(self):
        self.parse_F()
        self.parse_T_prime()

    def parse_T_prime(self):
        if self.current_token == '*':
            self.consume()
            self.parse_F()
            self.parse_T_prime()
        # else, it's ε (epsilon), do nothing

    def parse_F(self):
        if self.current_token == '(':
            self.consume()
```

```
            self.parse_E()
            self.match(')')
        elif self.current_token.isalpha():  # Assuming id is a single alphabet character
            self.consume()
        else:
            raise ValueError(f"Unexpected token: {self.current_token}")
# Example usage
input_string = "id+id*id$"
parser = Parser(input_string)
parser.parse()
```

Q3

CODE :-

```
import random
import string
def generate_password(first_name, last_name):
    initials = first_name[0].lower() + last_name[0].lower()
    uppercase_letter = random.choice(string.ascii_uppercase)
    numbers = ''.join(random.choices(string.digits, k=4))
    special_characters = ''.join(random.choices(string.punctuation, k=2))
    password = initials + uppercase_letter + numbers + special_characters

    # Shuffle the password to make it more secure
    password_list = list(password)
    random.shuffle(password_list)
    shuffled_password = ''.join(password_list)

    final_password = shuffled_password[:16]
 return final_password

# Example usage
first_name = "bilal"
last_name = "younas"
password = generate_password(first_name, last_name)
print("Generated Password:", password)
```

Q1

The regex library of C# is a set of classes and methods that provide regular expression functionality to .NET applications. It is part of the System.Text.RegularExpressions namespace.
It includes classes and methods for:
*Compiling regular expressions
*Matching regular expressions against text
*Extracting and manipulating text that matches regular expressions
*Replacing text that matches regular expressions
*Splitting text into parts based on regular expressions
* The regex library in C# is a powerful and flexible tool for working with regular expressions. It can be used to solve a wide variety of text processing and data validation problems.