ASSIGNMENT

1. How to implement precedence rules and associativity in java language. Give an example.

Operator precedence is implemented based on types of operators used in the expression
Operator precedence is priority of operators.
It's important to know the order of execution of operators. To determine this there are certain rules to be followed:

1. Braces : ( ) and [ ]
2. Increment and decrement operators : ++, --
3. Arithmetic operators : *, /, %
4. +, -
5. Relational operators : <, <=, >, >=, ==, !=
6. Boolean and Bitwise Operators : &, |, !, ~, ^, <<, >>, >>>
7. logical operators : &&, ||, !
8. Ternary operator : ?:
9. Assignment operators : =, +=, -=, *=, /=, %=

Associativity : if we have same order of precedence operators in expression, then which side onwards implementation starts is the Associativity. let us see rules for associativity :

1. ++, -- → left to right
2. +, -, ~, ! → right to left
3. *, /, % → left to right
4. +, - → left to right

5. <<, >>, >>> → left to right

6. <, >, <=, >=, instanceof → left to right

7. ==, != → left to right

8. & → left to right

9. ^ → left to right

10. | → left to right

11. && → left to right

12. || → left to right

13. ? : → right to left

14. =, +=, -=, *=, /=, %=, &=, ^=, |=, <<=, >>=, >>>= → left to right

<u>Example</u>:

$x = a - (++c) - (++b)$

step 1:

'=' associativity is right to left.

So, expression on left get assigned to 'x'.

step 2:

'( )' first priority

evaluate values in ( ) i.e., $x = a - (c+1) - (b+1)$

Step 3:

Now $x = a - c - 1 - b - 1$

step 4:

Implement ~~addition~~ subtraction operator

$x = a - b - c - 2$

If $a = 10, b = 5, c = 1$

$x = 2$

```
class Precedence{
public static void main
        (String args[]){
  int a=10,b=5,c=1,r;
  r= a- ++c - ++b;
  System.out.println(r);
 }
}
```

Output : 2

2. Design a class that represents a bank account and construct the methods to
   i) Assign initial values
   ii) Deposit an amount
   iii) withdraw amount after checking balance
   iv) Display name and balance. Do you need to use static keyword for above bank account program? Explain.

```
import java.io.*;
import java.util.Scanner;

~~class~~ ~~Bank~~
public class Bank Account
{
    public double deposit=0;
    public double withdraw=0;
    private double balance;

    BankAccount (double balance)
    {
        this.balance = balance;
    }

    Scanner sc = new Scanner(System.in);

    public void calculate()
    {
     System.out.println("Enter name of person:");
     String name = sc.nextLine();
     System.out.println(" 1. Deposit \n 2. Withdraw \n 3. Display");
     System.out.print(" Enter your option :");
     int choice = sc.nextInt();
```

19BQ1A05P4

```
switch (choice)
{
        case 1:
            System.out.println("You selected to deposit");
            System.out.print("Enter how much you want to deposit:");
            deposit= sc.nextDouble();
            balance+=deposit;
            System.out.println(" Balance is :"+balance);
            break;

       case 2:
            System.out.println("You selected to withdraw");
            System.out.print ("Checking your balance.....");
            System.out.println("Your balance amount is:" +
                                        balance);
            System.out.println (" Enter withdraw amount
                          less than balance :");
            withdraw = sc.nextDouble();
            balance = balance - withdraw;
            System.out.println("Balance is :"+balance);
            break;

       case 3:
            System.out.println("Name of accountholder is :"+name);
            System.out.println ("Your balance is:"+balance);
  }
}
public static void main(String args[])
{
    BankAccount  b = new BankAccount( }}315000);
     b calculate ();
   }
}
```

```
javac BankAccount.java
java BankAccount
Enter name of person: D
Enter your option: 3
```

No, There is no need to use static keyword in this bank account problem. In Java, static keyword is mainly used for memory management. Basically, static is used for a constant variable or method that is same for every instance of class. But in this program, either of balance, deposit, withdraw need not be constant throughout program for different accountholders. So, I conclude no need to use static keyword from my program.

3 Define a class Electric Bill with following specifications:
class : ElectricBill
Instance variable /data member:

String n - to store name of consumer
int units to store number of units consumed
double bill - to store amount to be paid.

Member methods:

void accept() — to accept name of consumer and no. of units.
void calculate() — to calculate bill as per following tarrif:

Number of units	— Rate per unit
First 100 units	— Rs 2.00
Next 200 units	— Rs 3.00
Above 300	— Rs 5.00

A surcharge of 25% charged if the number of units consumed is above 300 units.

void print() — To print details as follows:

19BQ1A05P4

Name of consumer: . . .

Number of units consumed: . . . .

Bill amount . . . . .

Write a main method to create an object of class and call above member methods.

```
import java.io.*;
import java.util.Scanner;
class ElectricBill
{
    Scanner sc= new Scanner(System.in);
    String n;
    int units;
    double bill;
    public void accept()
    {
        System.out.println("Enter consumer name:");
        n= sc.nextLine();
        System.out.println("Enter no. of units consumed:");
        units = sc.nextInt();
    }
    public void calculate()
    {
     if (units <100)
            bill = units * 2;
     else
      if (units>100 && units<300)
            bill = 100 *2 + (bill-100)* 3;
     else
       if (units >300)
            bill = 100*2 + 200 * 3 + (units - 300)*5;
            bill = bill + (2.5* bill) /100;
    }
public void print()
{
   System.out.println("Name of the consumer:"+n);
   System.out.println("Number of units consumed:"+units);
   System.out.println("Amount of bill to be paid:"+bill);
}
public static void main(string args[])
{
     ElectricBill e = new ElectricBill();
     e.accept();
     e.calculate();
     e.print();
  }
}
```

```
javac ElectricBill.java
java ElectricBill
Enter Consumer name:"5P4"
Enter no.of units : 280
Name of consumer: 5P4
Number of units : 280
Amount bill : 640
```

4. Design a class to overload a function check() as follows:

i) void check(String str, char ch) - to find and print the frequency of a character in a string.

Example:

Input - Output

Str = "success" number of s present in it is = 3

ch = 's'

ii) Void check (String s1) - to display only the vowels from String s1, after converting it to lower case.

Example:

Input:

s1 = "computer" output: o u e

```
import java io.*;
import java.util. Scanner;
class One
{
    public void check (String str, char ch)
    {
        int count = 0;
        for (int i=0; i<str.length( ); i++)
            if (ch == str.charAt(i))
                count++;
        System.out.println("Frequency of given char in
                                    String is = "+ count);
    }
}
class Two extends One
{
    public void check (String s1)
    {
        char[] vowels = {'a','e','i','o','u'};
        for (int i=0; i<s1.length() ; i++)
            for (int j=0; j<vowels.length ; j++)
                if (s1.charAt(i) == vowels[j])
                    System.out.print(Character.toLowerCase
                                        (vowels[j]+" "));
    }
}
public class Overload
{
    public static void main (String args[])
    { Two t = new Two();
```

```
        t.check("success", 's');

        t.check("computer");

    }
}
```

```
Javac Overload.java
java Overload
Frequency of given char in string
        is = 3

 o u e
```

In this program, I used Character wrapper class so that the char datatype convert to a object and it can be dereferenced.

* I used my textbook to clarify some doubts.