JAVA ASSIGNMENT
SET-2

Reg No:- 19BQ1A05E9

1) How to implement Precedence rules and associativity in java language? Give an example?

A) **JAVA OPERATORS PREFERENCE AND ASSOCIATIVITY:-**

Java operators have two properties i.e, precedence and Associativity. Precedence is the priority order of an operator, if there are two or more operators in an expression then the operator of highest priority will be excuted first. Then higher, later high. For example, in expression 1+2*5, multiplication(*) operator will be processed first and then addition(+). It because multiplication has higher priority (or) precedence than addition.

Alternatively, we can say that when an operand is shared by two operands. (2 in above example is shared by + and *) then higher priority operator picks the shared operand for processing. From above example, we can understand the role. when all operators in an expression have same priority, Associativity acts. It tells us the direction of execution of operators that can either be left to right or right to left. For example, in expression, a=b=c=8, the assignment operator is executed from right to left that is 'c' will be assigned by 8, then 'b' will be assigned by c. and finally will be assigned by 'b'. we can parenthesize as (a=(b=(c=8))) we can also change the priority of a java operator by enclosing the lower order priority operator in parenthesis but not the associativity. For example, In (1+2)*3, addition will be done first because paranthesis has higher priority than multiplication operator.

19BCA05EQ

Precedence	operator	Description	Associativity
1	[ ] ( ) .	array index method call member access	Left to Right
2	++ -- +- ~ !	pre/postfix increment pre/postfix decrement unary plus,minus bitwise NOT logical NOT	Right to left
3	(type cast) new	type cast object creation	Right to left
4	* / %	multiplication division modulus(remainder)	left to Right
5	+- +	addition, subtraction string concatenation	left to Right
6	<< >>>> >>>	left shift signed right shift unsigned or zero-fill right shift	left to Right
7	< <= >>= instanceof	less than less than or equal to greater than greater than or equal to reference test	left to Right
8	== !=	equal to not equal to	left to Right
9	&	bitwise AND	left to Right
10	^	bitwise XOR	left to Right
11	|	bitwise OR	left to Right
12	&&	logical AND	left to Right
13	?:	conditional(ternary)	Right to left
14	||	logical XOR	left to Right
15	= += -= *= /= %= &= ^= |= <<= >>= >>>=	assignment and short hand assignment operators	Right to left

2)
```
import java.util.Scanner;
Public class BankAccount {

    double act_num;
    String name;
    String act_type;
    int bal;
    void setData(double a, String b, String c, int d){

        act_num = a;
        name = b;
        act_type = c;
        bal = d;

    }
    void Despoit (int a){
        System.out.println("Balance before despoit is "+bal);

        bal = bal + a;
        System.out.println("Balance after deposit is "+bal);

    }
    void withdraw(int a){
        System.out.println("Balance before withdraw is "+bal);
        bal = bal - a;
        if(bal < 0){
            System.out.println("cannot withdraw");
            bal = bal - a;

        }
        else
            System.out.println("Balance after withdraw is "+bal);
```

19BQ1A05F9

```
Void Display() {
        System.out.Println("Name : " + name);
        system.out.Println("Balance: " + bal );
}
Public static void main(String[] args){
        Bank Account ba= new BankAccount();
        Scanner s= new scanner(system.in);
        System.out.println("Enter acc-num, name, type, bal");
        ba.set Data(s.nextInt(), s.next(), s.next(), s.nextInt());
        System.out.println("enter the amount to deposit");
        ba.Deposit(s.nextInt());
        system.out.println("enter the amount to withdraw");
        ba.withdraw(s.nextInt());
        ba.Display();
```

Output:

Enter acct-num, name, acct-type, bal

191059, Nagasai, savings, 30,000

Enter the amount to deposit

4000

Balance before Deposit is 30000

Balance after Deposit is 34000

Enter the amount to withdraw

10000

Balance before withdraw is 34000

Balance after withdraw is 24000

Name: Nagasai

Balance: 24000

19BQ1A05F9

19BQ1A05E9

Do you need to use static keyword for the above bank account program? Explain.

A) Static keyword:- The static keyword is used in java mainly for memory .. no object needs to be created to use static variable or call .static methods, just put the .. using a static variable we make our program memory efficient.

So, we no need to use the static keyword for the above bank account program.

3) Programme:-

```
import.java.Util.*;
Public class ElectricBill{
    int units;
    String n;
    double bill;
    Scanner ob = new Scanner(System.in);
    void accept(){
        System.out.println("Enter name of the customer");
        n = ob.next();
        System.out.println("Enter number of units consumed");
        units = ob.nextInt();
    }
    void calculate(){
        if (units<=100)
            bill = units*2;
        else
        if (units>100 && units<=300)
            bill = 100*2 + (units-300)*3;
```

19BQ1A05E9

4) Design a class to implement a Electric Bill [illegible]

19BQ1A05E9

A)

```
        else
            bill = 100*2 + 200*3 + (units - 300) * 5;

        if(units > 300)
            bill = bill + 2.5/100 * bill;
    }
    void Print() {
        System.out.println("Bill Amount: "+bill);
    }
    Public static void main(String args[]){
        Electric Bill obj = new Electric Bill();
        obj.accept();
        obj.calculate();
        Obj.Print();
    }
}
```

output-1:

Name of the customer:
M.V.Sai

No. of units consumed:
150

Bill Amount: 350.0

output-2:

Name of the customer:
M. Lakshmi

No. of units consumed:
375

Bill Amount: 1204.375

19BQ1A05E9

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

A) Programme:

```
Class overload{
    Public static void check (String s, char ch){
        int c=0;
        For (int i=0; i<s.length(); i++){
            if (s.charAt(i)==ch)
                c++;
        }
        System.out.println("number of "+ch+" Present is = "+c);
    }
    Public static void check (String s1){
        s1= s1.toLowerCase();
        For(int i=0; i<s1.length(); i++){
            char ch = s1.charAt(i);
            if (ch=='a'||ch=='e'||ch=='i'||ch=='o'||ch=='u')
                System.out.print(ch + " ");
        }
    }
    Public static void main (String args[]){
        check ("success", 's');
        check ("computer");
    }
}
```

OUTPUT:

number of s Present is = 3

o u e