

# JAVA Assignment-1

## Set-2

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

A) Order in which the operators in an expression are evaluated determines - the Operator

### Precedence.

In Java when an expression is evaluated, there may be more than one operators involved in an expression. When more than one operator has to be evaluated in an expression Java interpreter has to decide which operator should be evaluated first. Java makes this decision on the basis of the precedence and the associativity of the operators.

If an expression has two operators with similar precedence, the expression is evaluated according to its associativity (either left to right, or right to left).

### Example:

```
1. class Demo {  
    public static void main(String args[]) {  
        int myInt = 12 - 4 * 2;  
        System.out.println(myInt);  
    }  
}
```

output:

4 H.

According to the table below,

* Java Operator Precedence & Associativity *		
Operators	Precedence	Associativity
postfix increment & decrement	++, --	left to right
prefix increment & decrement, & unary	++, --, +, -, ~, !	right to left
multiplicative	*, /, %	left to right
additive	+, -	left to right
shift	<<, >>, >>>	left to right
relational	>, <, <=, >=, instance of	left to right
equality	==, !=	left to right
bitwise AND	&	left to right
bitwise exclusive OR	^	left to right
bitwise inclusive OR		left to right
logical AND	&&	left to right
logical OR		left to right
ternary	?:	left to right
assignment	=, +=, -=, *=, /=, %=, &=, ^=,  =, <<=, >>=, >>>=	right to left



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

Output:

2  
The operator precedence of prefix '++' is higher than that of '-'.

$$\begin{aligned} \text{result} &= a - ++c - ++b; \\ &= a - (++c) - (++b); \\ &= 10 - (1+1) - (5+5); \\ &= 10 - 2 - 6 \\ &= 10 - 8 \\ &= 2. \end{aligned}$$

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 the name and balance.

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

```
import java.io.*;
import java.util.Scanner;
public class BankAcc {
    String name;
    double balance;
    double depositnum;
    double withdrawnum;
    void assign(){
        Scanner read = new Scanner(System.in);
        System.out.println("Enter Account holder name:");
        name = read.nextLine();
        System.out.println("Enter Balance in the
                           account:");
        balance = read.nextDouble();
        System.out.println("Enter amount to be deposit:");
        depositnum = read.nextDouble();
        System.out.println("Enter amount to be
                           withdraw:");
        withdrawnum = read.nextDouble();
    }
    void deposit(){
        balance += depositnum;
    }
    void withdraw(){
        System.out.println("Balance in the Account
                           + balance);
        if (balance > withdrawnum){
            balance -= withdrawnum;
        }
        else{
            System.out.println("Withdrawing
                               Amount is not possible");
        }
    }
}
```



```
void print(){
    System.out.println("Name of the
                        AccountHolder: "+name);
    System.out.println("Balance in the
                        Account: "+balance);
}
public static void main(String args[]){
    BankAcc b = new BankAcc();
    b.assign();
    b.deposit();
    b.withdraw();
    b.print();
}
}
```

Input:

Divyasri

10000

5000

3000

Enter AccountHolder name:

Enter Balance in the account:

Enter amount to be deposit:

Enter amount to be withdraw:

Balance in the Account: 15000

Name of the AccountHolder: Divyasri

Balance in the Account: 12000

In Java, if we want to access class members, we must first create an instance of the class. But there will be situations where we want to access

class members without creating any variables. In those situations, we can use the static keyword in java.

So here, There is no static keyword used by me in the above Bank account program.

3. Define a class Electric Bill with the following specifications:

class : ElectricBill

Instance variable / data member:

String n - to store the name of the customer.

int units - to store the number of units consumed.

double bill - to store the amount to paid.

Member methods:

void accept() - to accept the name of the customer and number of units consumed.

void calculate() - to calculate the bill as per the following tariff:

no. of units - rate per unit

First 100 - Rs 2.00

next 200 - Rs 3.00

above 300 - Rs 5.00

A surcharge of 2.5% charged if the no. of units consumed is above 300 units.

void print() - to print the details as follows:

Name of the customer.....

No. of units consumed.....



Bill amount. . . .

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

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

public class ElectricBill {
    String n;
    int units;
    double bill;

    void accept() {
        Scanner reader = new Scanner(System.in);
        System.out.println("Name of the customer=");
        n = reader.nextLine();
        System.out.println("Number of units consumed=");
        units = reader.nextInt();
    }

    void calculate() {
        if (units <= 100)
            bill = 2.00 * units;
        if (units >= 100 && units <= 300)
            bill = 3.00 * units;
        if (units > 300) {
            bill = 5.00 * units;
            bill += bill * (2.5 / 100);
        }
    }

    void print() {
        System.out.println("Name of the customer:" + n);
        System.out.println("Number of units consumed:" + units);
        System.out.println("Bill amount:" + bill);
    }
}
```

```
public static void main(String args[]){  
    ElectricBill a = new ElectricBill();  
    a.accept();  
    a.calculate();  
    a.print();  
}
```

}

Input:

Name of the customer: divyasri  
Number of units consumed: 450

Output:

Name of the customer: divyasri  
Number of units consumed: 450  
Bill amount: 2306.25

4. Design a class to overload a function check() as,  
i) void check(String str, char ch) - to find & print the frequency of a character in a string.

Eg: str = "success" number of s present is = 3.  
ch = 's'.

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

Eg:

input: s1 = "computer"

output: oue



```

import java.io.*;
public class Design{
    private static void check(String str, char ch){
        int count=0;
        for(int i=0; i < str.length(); i++){
            if(str.charAt(i) == ch){
                count += 1;
            }
        }
        System.out.println("Number of " + ch + "
                           present is = " + count);
    }
    private static void check(String s1){
        s1 = s1.toLowerCase();
        for(int i=0; i < s1.length(); i++){
            if(s1.charAt(i) == 'a' || s1.charAt(i) == 'e' ||
               s1.charAt(i) == 'i' || s1.charAt(i) == 'o' ||
               s1.charAt(i) == 'u'){
                System.out.println("\t" + s1.charAt(i));
            }
        }
    }
    public static void main(String args[]){
        check("tummadivyasri", 'a');
        check("DIVYASRI");
    }
}

```

Output:

Number of a present is = 2

i a i

1. 1<sup>st</sup> question - resource from programiz.com.
2. remaining are all my own.

---

The End

---

Thank you