JAVA PROGRAMMING ASSIGNMENT-1

SET-2

19B01A05N2
CSE-D

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

Rules of precedence:

→ Precedence rules are used to determine which operator should be applied first if there are two operators with different precedence, and these follow each other in the expression.

→ In such case, the operator with the highest precedence is applied first.

Eg: 2+3*4 is evaluated as 2+(3*4) [with the result 14] since * has higher precedence than +.

Precedence and Associativity table:

precedence	operator	Description	Associativity
1	[ ] ( ) .	array index method call member access	Left → Right
2	++ -- + - ~ !	Pre or postfix increment pre or postfix decrement unary plus, minus bitwise NOT logical NOT	Right → Left
3	(type cast) new	type cast object creation	Right → Left
4.	* / %	Multiplication division modulus(remainder)	Left → Right
5.	+ - +	addition, subtraction String concatenation	Left → Right
6.	<< >> >>>	left shift Signed right shift unsigned or zero-fill right shift	Left → Right

Precedence	operator	Description	Associativity
7	< <= > >= instanceof	less than less than or equal to greater than greater than or equal to reference test	Left → Right
8	== !=	equal to not equal to	Left → Right
9	&	bitwise AND	Left → Right
10	^	bitwise XOR	Left → Right
11	|	bitwise OR	Left → Right
12	&&	logical AND	Left → Right
13	||	logical OR	Left → Right
14	?:	conditional (ternary)	Right → Left
15	= += -= *= /= %= &= ^= |= <<= >>= >>>=	assignment and short hand assignment operator	Right → Left

Example:

```
class precedence {
    public static void main(String[] args){
        System.out.println("Evaluating operator precedence with pre and post
                            addition unary operator");
        int a = 10, b = 5, c=1, result;
        result = a - ++c - ++b;
        System.out.print("Result is" +result);
    }
}
```

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.

Solution:

```
Class BankAccount
{
    //instance variable
    private String accountNum;
    private double balance;

    public BankAccount (String acctNum, double initialBalance)
    {
        accountNum=acctNum;
        balance = initialBalance;
    }
    public void deposit(double amount)  //note "mutator" method
    {
       double newBalance = balance+amount;
        balance =newBalance;                    //modified instance var
    }
    public void withdraw(double amount)
    {
        double newBalance =balance - amount;
        balance = newBalance;
    }
    public String getAccount()      //return value of instance var
    {                               //note "accessor" method
        return accountNum;
    }
```

```
    public String double getBalance()
    {
        return balance;
    }
    public void transferFundsA(BankAccount destination, double amount)
    {
        destination.balance = destination.balance + amount;
        this.balance = this.balance - amount;
    }
    public void transferFundsB(BankAccount destination, double amount)
    {
        destination.deposit(amount);
        this.withdraw(amount);
    }
}
public class BankAccountTest2
{
    public static void main(String[] args)
    {
        BankAccount first = new BankAccount("11111", 100000);
        BankAccount second = new BankAccount("222222", 50000);
        System.out.printf("Account #:%s has initial balance of $%.2f%n",
                first.getAccount(), first.getBalance());
        System.out.printf("Account #:%s has initial balance of $%.2f%n",
                second.getAccount(), second.getBalance());
        first.transferFundsA(second, 5000);
        System.out.println("\nAfter \"first.transferFunds(second, 5000)\"...");
        System.out.printf("Account #:%s has new balance of $%.2f%n",
                first.getAccount(), first.getBalance());
        System.out.printf("Account #:%s has new balance of $%.2f%n",
                second.getAccount(), second.getBalance());
```

```
Second.transferFundsB(first, 10000);
System.out.println("\nAfter \"second.transferFundsB(first, 10000)\" ...");
System.out.printf("Account #%s has new balance of $%.2f%n",
        first.getAccount(), first.getBalance());
System.out.printf("Account #%s has new balance of $%.2f%n",
        second.getAccount(), second.getBalance());
}
}
```

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 traffic/tariff:

Number of units - Rate per unit

First 100 units - Rs. 2.00

Next 200 units - Rs. 3.00

Above 300 units - Rs. 5.00

A surcharge of 2.5% charged if the number of units consumed is above 300 units

Void print() - To print the details as follows:

Name of the Customer - - - -

Number of units consumed - - - -

Bill amount - - - -

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

Solution:

```
import java.io.*;
class ElectricBill {
    String n;
    int units;
    double bill;
    Void accept() thrwos IOException {
        BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));
        System.out.print("Name of the customer = ");
        n=reader.readLine();
        System.out.print("Number of units consumed: ");
        String un= reader.readline();
        units = Integer.parseInt(un);
    }
    Void Calculate() {
        if(units<=100)
        bill =2.00*units;
        if (units<=300)
        bill = 3.00*units;
        if(units>300)
        bill = 5.00*units;
    }
    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[])throws IO Exception {
        ElectricBill eb =new ElectricBill();
        eb.accept();
        eb calculate();
    }
}
```

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 is :3

ch : 's'

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

Example:

Input;

Si = "compiler" output : oue

Solution:

```
class CharacterVowel {

        PublicVoid checkering (Str, char, ch) {

                int c=0, code, l, s;
                Str = str.toLowerCase();
                int len= str. length();
                for (code=97; code<122; code++)
                {
                    c=0;
                    for (i=0; i<len; i++){
                        ch = Str.charAt(i);
                        s=(int)ch;
                        if(s==code)
                         c=c+1;
                    }
                  ch=(char)code;
                  if(c!=0)
                  System.out.println("Frequency of "+ch+" is" +c);

            }

      }
```

```
public void check (String s1){
        int i;
        char ch=0,chr =0;
        for(i =0; i< s1.length(); i++) {
                ch = s1.charAt(i);
                if (Character.isUpperCase(ch))
                chr =Character.toLowerCase(ch);
                if ((s1.charAt(i) == 'a') || (s1.charAt(i)=='u'|| (s1.charAt(i)
                    == 'o') || (s1.charAt(i)== 'i')|| (s1.charAt(i)=='e'))
                System.out.println(s1.charAt(i));
        }
}
}
```