9/21/20

19BQ1A05L3
CSE-D
S. Leela Vardhan

JAVA

Assignment

SET-2

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

Java Operator Precedence :

Operator precedence determines the order in which the operators in an expression are evaluated when two operators share a common operand. The operator with the highest precedence is operated first.

Operator precedence table :

The table below lists the precedence of operators in java ; higher it appears in the table, the higher its precedence.

Operators	precedence
postfix increment and decrement	++ , --
prefix increment and decrement, unary	++ , -- , + , - , ~ , !
multiplicative	* , / , %
additive	+ , -
shift	<< , >> , >>>
relational	< , > , <= , >= , instanceof
Equality	== , !=
bitwise AND	&
bitwise exclusive OR	^
bitwise inclusive OR	!
logical AND	&&
logical OR	||
ternary	? :
assaignment	= , += , -= , *= , /=

Example:

```
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

Associativity of 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).

Precedence	Associativity
++, --	left to right
++, --, +, -, ~, !	right to left
*, /, %	left to right
+, -	left to right
<<, >>, >>>	left to right
<, >, <=, =>, instanceof	left to right
==, !=	left to right
&	left to right
^	left to right
|	left to right
&& , ||	left to right
?:	right to left

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.

Program:

```
import java.util.Scanner;
Public class Bankmain{
    Static float balance = 0.0f;
    float deposit;
    float withdraw;
    Static String name;
    Static Scanner sc = new Scanner(System.in);
    // methods
    Static void Deposit(float amount){
        balance += amount;
        System.out.println(amount + " deposited ");
        System.out.println("Current balance : "+ balance);
    }
    Static void Display(){
        System.out.println("name : "+ name);
        System.out.println("balance : "+ balance);
    }
    Static void Withdraw(float amount){
        if(balance < amount){
            System.out.println("not sufficient balance");
        }
```

```
        else {
            balance -= amount;
            System.out.println(amount + " withdrawn");
            System.out.println("balance: " + balance);
        }
    }
Public static void main(String args[]){
    System.out.println("enter initial values:");
    System.out.println("enter name:");
    name = sc.nextLine();
    System.out.println("enter balance:");
    balance = sc.nextFloat();
    System.out.println("enter 1: deposit, 2: withdraw,
                                        3: cur balance");
    int choice = sc.nextInt();
    if (choice == 1){
        System.out.println("enter amount to deposit:");
        float amount = sc.nextFloat();
        Deposit(amount);
    }
    else if (choice == 2){
        System.out.println("enter amount to withdraw:");
        float amount = sc.nextFloat();
        Withdraw(amount);
    }
    else if (choice == 3){
        System.out.println(Display());
    }
}
}
```

The code in the method is not dependent on instance creation and is not using any instance variable. A particular piece of code is to be shared by all the instance methods. The definition of the method should not be changed or overridden.

The static variable is a class level variable and it is common to all the class objects. Single copy of the static variable is shared among all the class objects. A static method manipulates the static variables in a class. It belongs to the class instead of the class objects and can be invoked without using a class object. The static initialization blocks can only initialize the static instance variables. These blocks are only excuted once when the class is loaded.

3. Define a class ElectricBill with the following specifications:

Class : ElectricBill

Instance variable / data members:

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 ...

Program:

```
import java.util.Scanner;
Public class ElectricBill {
    Public Static void main(String args[]){
        Bill eb = new Bill();
        eb.accept();
        eb.calculate();
        eb.print();
    }
}
class Bill{
    Static Scanner sc = new Scanner(System.in);
    Static String n;
    Static int units;
    Static double bill;
    Void accept(){
        System.out.println("Enter name and no of units");
        n = sc.nextLine();
        units = sc.nextInt();
    }
    Void calculate(){
```

```
        if(Units <= 100){
            bill = units * 2.0f;
        }
        else if (units > 100 && units <= 300){
            bill = units * 3.0f;
        }
        else{
            bill = units * 5.0f;
            float charge = ((bill/100)* 2.5f);
            bill += charge;
        }
    }
    void print(){
        System.out.println("name of customer" +n);
        System.out.println("no of units consumed "+units);
        System.out.println("Bill amount :" + bill);
    }
}
```

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 a present 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

Program :

```
import java.util.Scanner;
Public Class Check{
    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 ( count +" "+ ch + "found");
    }
    Static void check (String s1){
        s1 = s1.toLowerCase ();
        System.out.println ("the vowels in string are :");
        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.print(s1.charAt(i) + " ");
            }
        }
    }
Pubic static void main(String args[]){
    Scanner sc = new Scanner(System.in);
    String str = sc.nextLine();
    System.out.print("enter string str:");
    System.out.print("enter string s1:");
    String s1 = sc.nextLine();
    System.out.println("character to find:");
    char ch = sc.next().charAt(0);
    check(str, ch);
    check(s1);
}
}
```