

MID TERM ASSIGNMENT

ACADEMIC YEAR: 2020 TO 2021

Hall Ticket No. :

1	9	B	Q	1	A	0	S	L	S
---	---	---	---	---	---	---	---	---	---

Name of the Student : S. Kaarthikeya pavana kumar

Course : B-tech

Branch : ECE/CSE/EEE/IT

Subject : Java programming

ASSIGNMENT / MARKS DETAILS

To be filled by the Student			To be filled by the Subject Teacher		
Submission Date	Assignment	Signature of the Student	Max Marks	Marks Obtained	Signature of Subject Teacher
21-09-2020	1	S. Kaarthikeya	5		

INSTRUCTIONS TO THE STUDENTS

1. The assignment should be submitted to the subject teacher on or before the given schedule.
2. Answer should be written on both sides of the paper.

INSTRUCTIONS TO THE SUBJECT TEACHER

1. The Subject teacher has to value with red ball point pen only.
2. The Subject teacher should award the marks on the left hand side of the margin and at the end of the each answer.
3. Do not correct the marks by overwriting or by scratching and writing.
4. The Subject teacher has to post marks in the space provided.

VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY, NAMBUR
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Vision of the Department

To facilitate quality education by focusing on assimilation, generation and dissemination of knowledge in the area of Computer Science & Engineering to transform students into socially responsible engineers.

Mission of the Department

- Equip our graduates with the knowledge by *student centric teaching-learning process* and expertise to contribute significantly to the software industry and to continue to grow professionally.
- To train *socially responsible, disciplined engineers* who work with good leadership skills and can contribute for nation building.
- To make our graduates *aware of cutting edge technologies* and make them industry-ready engineers.
- To shape the department into a *centre of academic and research excellence*.

Program Educational Objectives

PEO-1	To provide the graduates with solid foundation in Computer Science and Engineering along with the fundamentals of Mathematics and Sciences with a view to impart in them high quality technical skills like modelling, analyzing, designing, programming and implementation with global competence.
PEO-2	To prepare and motivate graduates with recent technological developments related to core subjects like programming, databases, design of compilers and Network Security aspects and future technologies so as to contribute effectively for Research & Development by participating in professional activities like publishing and seeking copy rights.
PEO-3	To train graduates to choose an appropriate career in employment, higher education or entrepreneurship by empowering them to excel in competitive examinations, by preparing them for lifelong learning and by inculcating in them ethical leadership skills.
PEO-4	To train the graduates to have basic interpersonal skills and sense of social responsibility that paves them a way to become good team members and leaders.

JAVA PROGRAMMING

Assignment 1 (SET – 2)

Submitted to: Naga Sri Harsha sir

Submitted by: 19BQ1A05L5

SET 2

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

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.

3. Define a class Electric Bill with the following specifications: [15]

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 :

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.

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 s1) – to display only the vowels from string s1 , after converting it to lower case.

Example :

Input:

S1= "computer" output: o u e

1) Precedence rules :-

Precedence rules determines the order in which the operators in an expression are evaluated.

operator precedence table :-

The table below lists the precedence of operators in Java, higher it appears in the table, the higher it's precedence.

<u>Operators</u>	<u>Precedence</u>
Postfix increment and decrement.	++, --
Prefix increment and decrement, and unary	++, --, +, -, ~, !
Multiplicative.	*, /, %
additive	+, -
shift.	<<, >>, >>>
relational.	<, >, <=, >=, instance of
equality.	==, !=
bitwise AND	&
bitwise exclusive OR.	^
bitwise inclusive OR.	
logical AND	&&
logical OR.	
ternary	?:
Assignment.	=, +=, -=, *=, /=, %= &=, ^=, =, <<=, >>=, >>>=

Associativity of operators in Java:-

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).

The table below shows the associativity of Java operators along with their precedence.

<u>Operators</u>	<u>Precedence</u>	<u>Associativity</u>
Postfix increment and decrement.	$++$, $--$	left to right
Prefix increment and decrement and unary	$++$, $--$, $+$, $-$, \sim , $!$	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.	\wedge	left to right
Bitwise inclusive OR.	$ $	left to right
Logical AND	$\&\&$	left to right.
Logical OR.	$ $	left to right.
Ternary	$?:$	right to left.
Assignment.	$=$, $+=$, $-=$, $*=$, $/=$, $\%=$, $\&=$, $\wedge=$, $ =$, $<<=$, $>>=$, $>>>=$	left to right.

Example:- operator precedence.

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 '-' subtraction operator.

2) import java.util.Scanner;

Public class Bankmain { // Assigning Initial values from main

Static float balance = 0.0f; // declaring variables.

float deposit;

float withdraw;

Static String name;

Static Scanner SC = new Scanner(System.in);

Static void Deposit (float amount) { // Deposit amount

balance += amount;

System.out.println(amount + "deposited to account");

System.out.println("current balance: " + balance);

}

Static void Display() { // Display name and balance

System.out.println("Name: " + name);

System.out.println("Balance: " + balance);

}

```

static void Withdraw (float amount) {
    // withdraw amount after checking balance.
    if (balance < amount) {
        System.out.println("your balance is not sufficient");
    }
    else {
        balance -= amount;
        System.out.println(amount + "withdraws money");
        System.out.println("current balance: " + balance);
    }
}

```

```

Public Static void main (String[] args) { // Assigning values
    System.out.println("enter or assign initial values");
    System.out.println("enter name");
    name = Sc.nextLine();
    System.out.println("enter balance");
    balance = Sc.nextFloat();
    System.out.println("enter 1. deposit");
    System.out.println("                2. withdraw");
    System.out.println("                3. current balance");
    int choice = Sc.nextInt();
    if (choice == 1) { // Deposit amount
        System.out.println("enter amount to deposit");
        float amount = Sc.nextFloat();
        Deposit (amount);
    }
    else if (choice == 2) { // withdraw amount
        System.out.println("enter amount to withdraw");
        float amount = Sc.nextFloat();
        withdraw (amount);
    }
}

```



```

    else if (choice == 3) {
        display();
    }
    else {
        System.out.println("Choice + "Entered valid input");
    }
}
}
}

```

Yes, we need to use static keyword for the above bank account program.

In Java, static keyword is mainly used for memory management. It can be used with variables, methods, blocks and nested classes. It is a keyword which is used to share the same variable or method of a given class. Basically, static is used for a constant variable or a method that is same for every instance of class.

```

3) import java.util.Scanner;

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

class Bill {
    static Scanner sc = new Scanner(System.in);
    static String n; // to store name of customer.
    static int units; // to store number of units consumed.
    static double bill; // to store the amount to paid.
}

```

```
void accept() { // to accept name and units consumed.  
    System.out.println("enter name:");  
    n = sc.nextLine();  
    System.out.println("enter no of units");  
    units = sc.nextInt();  
}
```

}

```
void calculate() { // to calculate the bill.
```

```
    if(units <= 100) {
```

```
        bill = units * 2.00f;
```

}

```
    else if (units > 100 && units <= 300) {
```

```
        bill = units * 3.00f;
```

}

```
    else {
```

```
        bill = units * 5.00f;
```

```
        float charge = ((bill/100) * 2.5f);
```

```
        bill += charge;
```

}

}

```
void print() { // to print the details.
```

```
    System.out.println("Name of the customer:" + n);
```

```
    System.out.println("Number of units consumed:" + units);
```

```
    System.out.println("Bill amount:" + bill);
```

}

}

4.) Public class check {

static void check (String str, char ch) {

// to find and print the frequency of character in string

int count = 0;

for (int i = 0; i < str.length(); i++) {

if (str.charAt(i) == ch) {

count = count + 1;

}

}

system.out.print("no of " + ch);

system.out.print(" in string are: " + count);

}

static void check (String s1) { // to display vowels in s1

s1 = s1.toLowerCase();

system.out.println("the vowels in the 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) + " ");

}

}

}

public static void main (String[] args) {

/* String str = "success"

String s1 = "computer"

char ch = 's'

As Given in example */


```
Scanner SC = new Scanner(System.in);  
str = SC.nextLine();  
s1 = SC.nextLine();  
ch = SC.next();  
check(str, ch);  
check(s1);
```

```
}
```

```
}
```

Example :-

Input - str = "success"
 ch = 's'
 s1 = "computer"

output - no of s in string are : 3

the vowels in the string are : o u e.