JAVA PROGRAMMING ASSIGNMENT

Name ÷ Ponnekanti Lokesh

Roll No ÷ 19BQ1A05I1

Sec ÷ CSE-C

1) Precedence rules and associarity in java

Operator precedence determines the grouping of terms in an expression. This affects how an expression is evaluated Certain operators have higher precedence than others.

Eg: y = 1 + 2 * 3

Here y is assigned as 7, but not 9 because multiplication operator has higher precedence than addition.

Here, according the table, the operators will be evaluated in an expression.

operator.	Precedence	Associativity
Postfix	>() [] .	Left to right
unary	> ++ -- ! ~	Right to left
Multiplicative	>*/	Left to Right
Additive	>+-	Left to Right
shift	>>> <<	Left to Right
Relational	>> >= <=	Left to Right
Equality	>== !=	Left to Right
Bitwise	>&(1) >^(2) >|(3)	Left to Right
Logical AND	>&&	Left to Right
Logical OR	>||	Left to Right
conditional	?:;	Right to left
Asiginment	>+ >= = *= /= >>=	Right to Left

Implementation

In an priority queue, an element with high priority is served before an element with low priority. If two elements with the same priority, they are served according to their order in the queue.

Eg:-

```
class Myclass {
public static void main (string [] args) {
    int a = 1
    int b = 2
    int c = 3
    System.out.println (" a + b = " + a + b);
    System.out.println (" a + b = " + (a + b));
    System.out.println (" c = " + a + b);
    System.out.println (" c = " + (a + b));
```

O/P :-

```
a + b = 12
a + b = 3
c = 12
c = 3
```

Here, both of the operands of + operator are same the other is automatically cast to string. String concantetion & addition have same precedence. But paranthesis in 2 and 4 statements + operation performed instead of string concatenation.

```
public class Bankaccount
{
    private string name ;
    private double balance;

    Public Bank account ( )
    {
        name = "    ";
        balance = 0;
    }
    Public Bankaccount (string name , double balance)
    {
        name = "    ";
        balance = initial balance ;
    }
    public void deposit (double amount)
    {
        balance = balance + amount}
    }
    public void withdraw (double amount)
    {
        balance = balance - amount;
    }
    Public double getBalane ( )
    {
        return balance;
    }
```

```
Public class ~~Savin~~ Bankaccount
{
    public static void main (String[] args)
    {
        Bankaccount sc = new Bankaccount ("Sai", 1000);
        sc.withdraw(250);
        sc.deposit (400);
        System.out.println (~~jimmy~~ sc.getBalance());
    }
}
```

o/p :-

1150.

```
import java.util.scanner;
import java.io.*;
class ElectricBill{
 string n;
 int units;
 double bill;
 void accept()
 Scanner sc = new scanner (system.in);
 System.out.println("Name of the customer=");
 n= reader.readline();
 system.out.println("Nomber of units consomed=");
 String n = reader.readline();
 units = Integer.parseInt(n);
}
Void print(){
 System.out.println("Name of the customer = "+n);
 system.
 Void caluclate{
 if (units <= 100)
 bill = 2.00*units;
 if (units <=300 && units >=100)
 bill= 3.00*units;
```

```
if (units > 300){
 bill= 5.00*units
 bill = bill *(2.5/100)
}
Void print(){
 System.out.println("Name & customer:"+n);
 System.out.println("Name & units consumed:"+units);
 system.out.println("Bill amount:"+bill);
}
```

```
class charVowel;
public void check(str, char ch){
int c=0, code, i, 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)=='e') ||
     (s1.charAt(i)=='o') || (s1.charAt(i)=='u') ||
     (s1.charAt(i)=='i'))
system.out.println(s1.charAt(i));
 }
}
}
```

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.