

Q15:- Define a class **WordExample** having the following description:

Data members/instance variables: private String strdata : to store a sentence.

Parameterized Constructor WordExample(String) : Accept a sentence which may be terminated by either '.', '? 'or '!' only. The words may be separated by more than one blank space and are in UPPER CASE.

Member Methods: void countWord(): Find the number of words beginning and ending with a vowel.

void placeWord(): Place the words which begin and end with a vowel at the beginning, followed by the remaining words as they occur in the sentence.

Code:-

```
import java.util.*;

class WordExample
{
    private String strdata;

    public WordExample(String str)
    {
        strdata=str;
    }

    public void wordCount()
    {
        int i,ln,wln,c1,c2,c=0;

        String tmp;

        char ch1,ch2;

        ln=strdata.length();

        for(i=0;i<ln;i++)
        {
            tmp="";

            c1=0;

            c2=0;

            for(;strdata.charAt(i)!=' ';i++)
            {
                if(i==(ln-1))
                {
```

```

        break;
    }
    tmp=tmp+strdata.charAt(i);
}
if(!tmp.isEmpty())
{
    wln=tmp.length();
    ch1=tmp.charAt(0);
    ch2=tmp.charAt(wln-1);
    switch(ch1)
    {
        case 'A':
        case 'E':
        case 'I':
        case 'O':
        case 'U': c1++;
    }
    switch(ch2)
    {
        case 'A':
        case 'E':
        case 'I':
        case 'O':
        case 'U': c2++;
    }
    if(c1==1&& c2==1)
        c++;
    }
}
System.out.println("Number Of Words Are= "+c);
}

```

```

public void placeWord()
{
    int i,ln,wln,c1,c2,c=0;
    String tmp,fstr="",lstr="";
    char ch1,ch2;
    ln=strdata.length();
    for(i=0;i<ln;i++)
    {
        tmp="";
        c1=0;
        c2=0;
        for(;strdata.charAt(i)!=' ';i++)
        {
            if(i==(ln-1))
            {
                break;
            }
            tmp=tmp+strdata.charAt(i);
        }
        if(!tmp.isEmpty())
        {
            wln=tmp.length();
            ch1=tmp.charAt(0);
            ch2=tmp.charAt(wln-1);
            switch(ch1)
            {
                case 'A':
                case 'E':
                case 'I':
                case 'O':
                case 'U': c1++;
            }
        }
    }
}

```

```

    }
    switch(ch2)
    {
        case 'A':
        case 'E':
        case 'I':
        case 'O':
        case 'U': c2++;
    }
    if(c1==1&& c2==1)
        fstr=fstr+tmp+" ";
    else
        lstr=lstr+tmp+" ";
    }
}
tmp=fstr+lstr;
tmp=tmp.trim();
tmp=tmp+".";
System.out.println(tmp);
}
public static void main(String []args)
{
    Scanner sr=new Scanner(System.in);
    System.out.print("Sentence: ");
    String s = sr.nextLine();
    s=s.trim();
    s=s.toUpperCase();
    int len = s.length();
    char last = s.charAt(len - 1);
    if(last != '.' && last != '?' && last != '!')
    {

```

```
        System.out.println("INVALID INPUT");  
        System.exit(0);  
    }  
    WordExample ob=new WordExample(s);  
    ob.wordCount();  
    ob.placeWord();  
}  
}
```

Output:-

```
C:\Users\Gaurav\Desktop\Java Programs>javac WordExample.java  
  
C:\Users\Gaurav\Desktop\Java Programs>java WordExample  
Sentence: How are you?  
Number Of Words Are= 1  
ARE HOW YOU.  
  
C:\Users\Gaurav\Desktop\Java Programs>|
```

Q16:- Write a Java program to create a class called **ArrayDemo** and overload arrayFunc() function. **void arrayFunc(int [], int) →** To find all pairs of elements in an Array whose sum is equal to a given number :

Array numbers= [4, 6, 5, -10, 8, 5, 20], target=10

void arrayFunc(int A[], int p, int B[], int q) → Given two sorted arrays A and B of size p and q, Overload method arrayFunc() to merge elements of A with B by maintaining the sorted order i.e. fill A with first p smallest elements and fill B with remaining elements.

Code:-

```
import java.util.*;
class ArrayDemo
{
    public void arrayFunc(int arr[],int sum)
    {
        System.out.println("Pairs of elements whose sum is "+sum+" are : ");
        for (int i = 0; i < arr.length; i++)
        {
            for (int j = i+1; j < arr.length; j++)
            {
                if(arr[i]+arr[j]==sum)
                {
                    System.out.println(arr[i]+" + "+arr[j]+" = "+sum);
                }
            }
        }
    }
    public void arrayFunc(int[] A, int p, int[] B, int q)
    {
        for (int i = 0; i < p; i++)
        {
            if (A[i] > B[0])
            {
                int temp = A[i];
                A[i] = B[0];
                B[0] = temp;
                int first_arr = B[0];
                int k;
                for (k = 1; k < q && B[k] < first_arr; k++)
                {
                    B[k - 1] = B[k];
                }
                B[k - 1] = first_arr;
            }
        }
    }
}
```

```

    }
}
public class MethodOver
{
    public static void main(String args[])
    {
        ArrayDemo ob=new ArrayDemo();
        int[] arr = { 1, 5, 7, -1, 5 };
        int sum = 6;
        ob.arrayFunc(arr, sum);
        int[] A = { 1, 5, 6, 7, 8, 10 };
        int[] B = { 2, 4, 9 };
        int p = A.length;
        int q = B.length;
        System.out.println("\nOriginal Arrays:");
        System.out.println("A: " + Arrays.toString(A));
        System.out.println("B: " + Arrays.toString(B));
        ob.arrayFunc(A, p, B, q);
        System.out.println("\nSorted Arrays:");
        System.out.println("A: " + Arrays.toString(A));
        System.out.println("B: " + Arrays.toString(B));
    }
}

```

Output:-

```

C:\Users\Gaurav\Desktop\Java Programs>javac MethodOver.java

C:\Users\Gaurav\Desktop\Java Programs>java MethodOver
Pairs of elements whose sum is 6 are :
1 + 5 = 6
1 + 5 = 6
7 + -1 = 6

Original Arrays:
A: [1, 5, 6, 7, 8, 10]
B: [2, 4, 9]

Sorted Arrays:
A: [1, 2, 4, 5, 6, 7]
B: [8, 9, 10]

```

Q17:- Write a java program to calculate the area of a rectangle, a square and a circle. Create an abstract class '**Shape**' with three abstract methods namely **rectangleArea()** taking two parameters, **squareArea()** and **circleArea()** taking one parameter each. Now create another class '**Area**' containing all the three methods **rectangleArea()**, **squareArea()** and **circleArea()** for printing the area of rectangle, square and circle respectively. Create an object of class **Area** and call all the three methods.

Code:-

```
abstract class Shape
{
    abstract double rectangleArea(double len,double br);
    abstract double squareArea(double side);
    abstract double circleArea(double radius);
}
class Area extends Shape
{
    double rectangleArea(double length,double breadth)
    {
        return length * breadth;
    }
    double squareArea(double side)
    {
        return side * side;
    }
    double circleArea(double radius)
    {
        return (22.0/7.0) * radius * radius;
    }
}
public class CalArea
{
    public static void main(String arg[])
    {
        Shape sp = new Area();
        System.out.println("Rectangle Area : " + sp.rectangleArea(10,4));
        System.out.println("Square Area : " + sp.squareArea(7));
        System.out.println("Circle Area : " + sp.circleArea(3.5));
        System.out.println();
    }
}
```

Output:-

```
C:\Users\Gaurav\Desktop\Java Programs>javac CalArea.java
```

```
C:\Users\Gaurav\Desktop\Java Programs>java CalArea
Rectangle Area : 40.0
Square Area : 49.0
Circle Area : 38.5
```


Q18:- Write a java program to implement abstract class and abstract method with following details:

Create a abstract Base Class Temperature

Data members: double temp;

Method members: void setTempData(double)
abstract void changeTemp()

Sub Class Fahrenheit (subclass of Temperature)

Data members: double ctemp;

method member: Override abstract method changeTemp() to convert Fahrenheit temperature into degree Celsius by using formula $C = 5/9 * (F - 32)$ and display converted temperature

Sub Class Celsius (subclass of Temperature)

Data member: double ftemp;

Method member: Override abstract method changeTemp() to convert degree Celsius into Fahrenheit temperature by using formula $F = 9/5 * c + 32$ and display converted temperature.

Code:-

```
abstract class Temperature
{
    protected double temp;
    void setTempData(double tmp)
    {
        temp=tmp;
    }
    abstract void changeTemp();
}
class Fahrenheit extends Temperature
{
    double ctemp;
    void changeTemp()
    {
        ctemp=5.0/9.0*(temp-32.0);
        System.out.println("Fahrenheit Into Degree Celsius is=> "+ctemp);
    }
}
class Celsius extends Temperature
{
    double ftemp;
    void changeTemp()
    {
        ftemp=(9.0/5.0)*temp+32.0 ;
        System.out.println("Degree Celsius into Fahrenheit is=> "+ftemp);
    }
}
public class ConvertTemp
{
    public static void main(String [] args)
    {
        Temperature ref;
```

```
        Fahrenheit fr=new Fahrenheit();  
        fr.setTempData(104);  
        Celsius cr=new Celsius();  
        cr.setTempData(40);  
        ref=fr;  
        ref.changeTemp();  
        ref=cr;  
        ref.changeTemp();  
    }  
}
```

Output:-

```
C:\Users\Gaurav\Desktop\Java Programs>javac ConvertTemp.java
```

```
C:\Users\Gaurav\Desktop\Java Programs>java ConvertTemp  
Fahrenheit Into Degree Celsius is=> 40.0  
Degree Celsius into Fahrenheit is=> 104.0
```

Q19:- Write a java program to create an interface that consists of a method to display **volume()** as an abstract method and redefine this method in the derived classes to suit their requirements. Create classes called **Cone**, **Hemisphere** and **Cylinder** that implements the interface. Using these three classes, design a program that will accept dimensions of a cone, cylinder and hemisphere interactively and display the volumes.

Volume of cone = $(1/3)\pi r^2 h$

Volume of hemisphere = $(2/3)\pi r^3$

Volume of cylinder = $\pi r^2 h$

Code:-

```
interface CalVolume
{
    static double pi = 3.14159;
    void volume();
}
class Cone implements CalVolume
{
    double r;
    double h;;
    Cone(double r,double h)
    {
        this.r=r;
        this.h=h;
    }
    public void volume()
    {
        double vol;
        vol=(double)1/3*pi*h*r*r;
        System.out.println("Volume of Cone is=> "+vol);
    }
}
class Hemisphere implements CalVolume
{
    double r;
    Hemisphere(double r)
    {
        this.r=r;
    }
    public void volume()
    {
        double vol;
        vol=(double)(2*pi*(double)Math.pow(r,3))/(double)(3);
        System.out.println("Volume of Hemisphere is=> "+vol);
    }
}
class Cylinder implements CalVolume
{
    double r;
    double h;;
```

```

    Cylinder(double r,double h)
    {
        this.r=r;
        this.h=h;
    }
    public void volume()
    {
        double vol;
        vol = pi*r*r*h;
        System.out.println( "Volume of Cylinder is => "+vol );
    }
}
public class VolumeCal
{
    public static void main(String args[])
    {
        Cone c1=new Cone(5.0,12.0);
        Hemisphere h1=new Hemisphere(11.0);
        Cylinder cy=new Cylinder(5.0,7.0);
        CalVolume ref;
        ref=c1;
        ref.volume();
        ref=h1;
        ref.volume();
        ref=cy;
        ref.volume();
    }
}

```

Output:-

```

C:\Users\Gaurav\Desktop\Java Programs>javac VolumeCal.java

C:\Users\Gaurav\Desktop\Java Programs>java VolumeCal
Volume of Cone is=> 314.159
Volume of Hemisphere is=> 2787.6375266666667
Volume of Cylinder is => 549.77825

```