Create a class student with following details
Data members : name, roll, marks[]
Parameterised constructor
double avg() : to return the average of marks
void display()
create 2 objects and print the details

```
import java.util.Scanner;

public class student {

 String name;
 int roll;
 double[] marks;

 student(String nam, int r, double m[]){
 name=nam;
 roll=r;
 marks=m;
 System.out.println("Student created");
 }

 double avg() {
 int l=marks.length;
 double avrg=0.0;
 for(int i=0;i<l;i++)
 avrg += marks[i];
 avrg = avrg/l;
 return avrg;
 }

 void display(){
 System.out.println("Name : "+name);
 System.out.println("Roll no : "+roll);
 System.out.println("Avg marks : "+ avg());
 }
```

```
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String nam;
        System.out.println("Enter name");
        nam = sc. nextLine();
        student s1 = new student(nam, 1, new double[]{86,
92, 59});
        student s2 = new student("Aman", 2, new double[]
{51, 65.5, 75});

        s1.display();
        s2.display();
    }

}
```

Create a class Calc with the following details
int add(int a, int b)
double add(int a, double b)
double add(double a, int b)
double add(double a, double b)

create a main method, call the methods to demonstrate method overloading.

```
public class calc {

    int add(int a, int b){
        System.out.println("From method 1");
        return a+b;
    }
```

```
    double add(int a, double b){
        System.out.println("From method 2");
        return a+b;
    }

    double add(double a, int b){
        System.out.println("From method 3");
        return a+b;
    }

    double add(double a, double b){
        System.out.println("From method 4");
        return a+b;
    }

    public static void main(String[] args) {
        calc c = new calc();
        System.out.println(c.add(2,3));
        System.out.println(c.add(2,3.7));
        System.out.println(c.add(2.5,3));
        System.out.println(c.add(2.5,3.7));
    }
}
```

Create a parent class animals
Create 2 child classes dog and lion inheriting animals
Define their common behaviors in the parent class
For the uncommon behaviours use abstract methods and
define later in the subclass

```
public abstract class Animal {

 String name;
 abstract void makeSound();

 public void walk(){
 System.out.println(name +" is walking");
 }
}



class dog extends Animal{

 dog(){
 name = "Dog";
 }
 @Override
 void makeSound() {
 System.out.println(name +" is barking");
 }
}

class lion extends Animal{

 lion(){
 name = "Lion";
 }
 @Override
 void makeSound() {
 System.out.println(name +" is roaring");
 }
}
```

```
class zoo{
    public static void main(String[] args) {
        dog d = new dog();
        lion l=new lion();

        d.walk();
        d.makeSound();

        l.walk();
        l.makeSound();
    }
}
```

Write a program to input a sentence, print the number of words in it. Also print the words which have even number of characters.

```
import java.util.Scanner;

public class Prac1 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter a sentence");
        String s = sc.nextLine();
        String[] words = s.split(" ");
        String word;
        int l = words.length;
        System.out.println("The number of words : "+l);
        for (int i =0; i<l; i++){
            word = words[i];
            if(word.length()%2==0){
                System.out.println(word);
```

```
        }
      }
   }
}
```

Write a program to input a word and capitalise every alternate character. Print the new word.

```
import java.util.Scanner;

public class Cap {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter a word");
        String word = sc.next();
        String newWord = "";
        int l = word.length();
        for(int i=0; i<l; i++){
            if(i%2==0)

newWord+=Character.toUpperCase(word.charAt(i));
            else

newWord+=Character.toLowerCase(word.charAt(i));
        }
        System.out.println(newWord);

    }

}
```