INHERITANCE CONCEPT

WRITE A JAVA PROGRAM TO CALCULATE SUM OF SERIES AND FACTORIAL

```
import java.util.*;
class Sumofseries
{
   protected int n;
   protected int i;
   private int sum=0;
   void get()
   {
      System.out.println("Enter a number:");
      Scanner sc=new Scanner(System.in);
      n=sc.nextInt();
   }
   void cal()
   {
      for(i=0;i<=n;i++)
      {
         sum=sum+i;
      }
      System.out.println(sum);
   }
}
class Factorial extends Sumofseries
{
   protected int fact=1;
   void cal()
   {
      super.cal();
      for(i=1;i<=n;i++)
      {
         fact=fact*i;
      }
      System.out.println(fact);
   }
}
class inh
{
   public static void main(String args[])
   {
      Factorial obj=new Factorial();
      obj.get();
```

```
        obj.cal();
    }
}
```

```
Enter a number:
5
15
120
```

WRITE A JAVA PROGRAM TO CALCULATE SUM OF SERIES AND FACTORIAL USING CONSTRUCTOR

```
import java.util.*;
class Sumofseries
{
    protected int n;
    protected int i;
    private int sum=0;
    Sumofseries()
    {
        System.out.println("Enter a number:");
        Scanner sc=new Scanner(System.in);
        n=sc.nextInt();
    }
    void cal()
    {
        for(i=0;i<=n;i++)
        {
            sum=sum+i;
        }
        System.out.println(sum);
    }
}
class Factorial extends Sumofseries
{
    protected int fact=1;
    void cal()
    {
        super.cal();
        for(i=1;i<=n;i++)
        {
            fact=fact*i;
        }
```

```
        System.out.println(fact);
    }
}
class inh
{
    public static void main(String args[])
    {
        Factorial obj=new Factorial();
        obj.cal();
    }
}
```

WRITE A JAVA PROGRAM USING INHERITANCE VALUES PASSING THROUGH ARGUMENTS

```
class test
{
    protected int a;

    void add(int a)
    {

        System.out.println(a);
    }
}

class addition extends test
{
    protected int b;
    void add(int a,int b)
    {
        this.a=a;
        this.b=b;
        super.add(a);
        int c = a + b;
        System.out.println(c);
    }
}

class inh
{
    public static void main(String args[])
    {
```

```
        addition obj = new addition();
        obj.add(2, 3);
    }
}
```

STRING MANIPULATION

```
class test
{
    String string1;
    String string2;
    String string3;
    String s1;
    String s2;
    String s3;
    String s4;
    String s5;
    int s6;
    int s7;
    boolean s8;
    boolean s9;
    char s10;
    int s11;

    void input()
    {
        string1 = "I Love java";
        string2="JAVA";
        string3="java";
        s1=string1.toLowerCase();
        s2=string1.toUpperCase();
        s3=string1.replace("o","i");
        s4=string1.substring(2);
        s5=string1.substring(2,6);
        s6=string1.indexOf("L");
        s7=string1.length();
        s8=string2.equals(string3);
        s9=string2.equalsIgnoreCase(string3);
        s10=string1.charAt(5);
        s11=string2.compareTo(string3);

    }
```

```
    void disp()
    {
        System.out.println(s1);//print s1 as i love java
        System.out.println(s2); // print s1 has I LOVE JAVA
        System.out.println(s3); //print I Live java
        System.out.println(s4);  //Display Love java
        System.out.println(s5); // Display Love
        System.out.println(s6);  // Find the position of L in s1
        System.out.println(s7);
        System.out.println(s8);
        System.out.println(s9);
        System.out.println(s10);
        System.out.println(s11);
    }

    public static void main(String args[])
    {
        test obj = new test();
        obj.input();
        obj.disp();
    }
}
```

OUTPUT:

```
i love java
I LOVE JAVA
I Live java
Love java
Love
2
11
false
true
e
-32
```

```
import java.util.*;
class username
{
    String str1,str2;
```

```
    void input()
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter username:");
        str1=sc.nextLine();
        System.out.println("Reenter username::");
        str2=sc.nextLine();
    }
    void disp()
    {
        if(str1.equals(str2))
        {
            System.out.println("Valid username");
        }
        else
        {
            System.out.println("Invalid username");
        }
    }
    public static void main(String args[])
    {
        username obj= new username();
        obj.input();
        obj.disp();

    }
}
```

OUTPUT:

```
Enter username:
saveetha@123
Reenter username::
savetha@456
Invalid username
```

Assignment4 (07-10-‘23)

1.Write a java program to read a character until a * is encountered. Also count the number of uppercase, lowercase, and numbers entered by the users.

Sample Input:

```
Enter * to exit…
Enter any character: W
Enter any character: d
Enter any character: A
```

Enter any character: G
Enter any character: g
Enter any character: H
Enter any character: *
Sample Output:
Total count of lower case:2
Total count of upper case:4
Total count of numbers =0
Test cases:
1. 1,7,6,9,5
2. S, Q, I, K,7, j, M
3. M, j, L, &, @, G
4. D, K, I, 6, L, *
5. *, K, A, e, 1, 8, %, *

```
import java.util.Scanner;

public class CharacterCount {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        int upperCount = 0;
        int lowerCount = 0;
        int numberCount = 0;

        char input;

        System.out.println("Enter * to exit...");

        do {
            System.out.print("Enter any character: ");
            input = scanner.next().charAt(0);

            if (Character.isUpperCase(input)) {
                upperCount++;
            } else if (Character.isLowerCase(input)) {
                lowerCount++;
            } else if (Character.isDigit(input)) {
                numberCount++;
            }
        } while (input != '*');

        System.out.println("Total count of lower case: " + lowerCount);
        System.out.println("Total count of upper case: " + upperCount);
```

```
        System.out.println("Total count of numbers: " + numberCount);


    }
}
```

2.Bring out the situation in which member names of a subclass hide members by the same name in the super class. How it can be resolved? Write Suitable code in Java and Implement above scenario with the Parametrized Constructor (accept int type parameter) of the Super Class can be called from Sub Class Using super () and display the input values provided.

Input :

Assign or input values for super class and sub class members.

Pseudo :

Define super class and sub class with one member (has same name)
Define method in super class and sub class with same method signature
Declare the object in main method
Invoke methods using object to display the values

Output :
Sample Input : 100, 200
Sample Output : 100, 200
Test Cases
1. 10, 20
2. -20, -30
3. 0, 0
4. EIGHT FIVE
5. 10.57, 12.58

```
class SuperClass {
    int value;
    int value1;

    SuperClass(int value) {
        this.value = value;
    }

    void display() {
        System.out.println(value);
    }
}
```

```
class Superclass extends SuperClass {



    Superclass(int value, int value1) {
        super(value);
        this.value1 = value1;
    }

    void display() {
        super.display();
        System.out.println(value1);
    }
}

class Main {
    public static void main(String[] args) {

        Superclass obj1 = new Superclass(100, 200);

        obj1.display();
    }
}
```

OUTPUT:

```
100
200
```