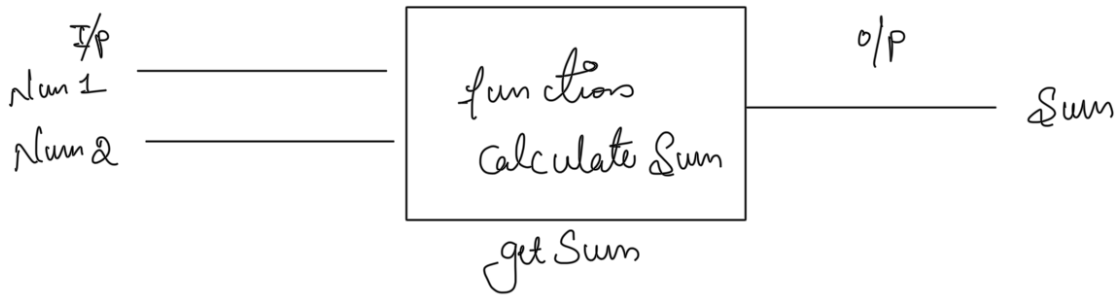
Hands on practice

① Function to get Sum of Numbers



int getSum(num1, num2){

o/p return Type

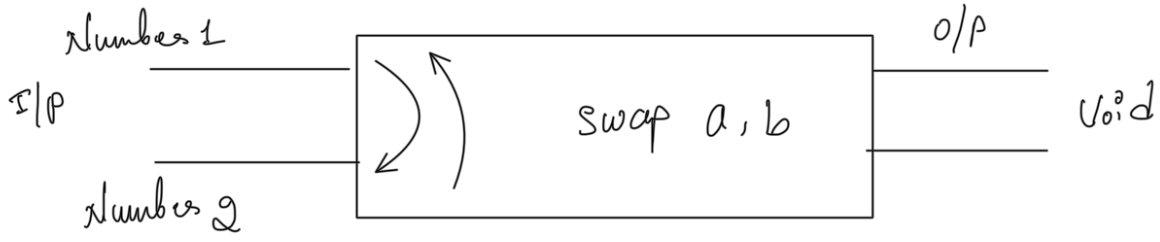
return (num1 + num2)

}

```
Coding > Java > Functions > Practice.java > Java Language Support > Practice > invoke_getSum
1   package Functions;
2
3   public class Practice {
4
5       public static int getSum(int num1,int num2){
6           return num1+num2;
7       }
8       public static void invoke_getSum(){
9           System.out.println("Invoking getSum(4,6): Result =  "+getSum(num1:4,num2:6));
10
11      }
        Run | Debug | Run main | Debug main
12      public static void main(String[] args) {
13          invoke_getSum();
14      }
15  }
16
```

---

② Function to swap two variables value

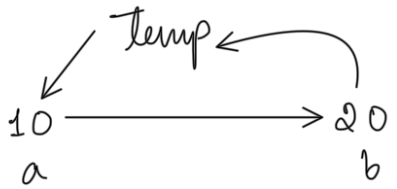


Equation for swapping Two Variable

int a=5 ; int b=10 ;

```
a = a + b - (b = a);
```

Swap using temp Variable




```
temp = a ;
a = b ;
b = temp ;
```

NOTE: Java pass Value to the function arguments Not the Actual reference.

Value passing By Value.

Eighter we have to create object or have to store them in the Array.

This doesn't Swap the Variables.

```
}

public static void swap(int num1 , int num2){
    int temp = num1;
    num1 = num2;
    num2 = temp;
}

public static void invoke_swap(){
System.out.println(x:"Invoking Swap(10,20)");
    int a=10;
    int b=20;
    System.out.println("Before Swap "+ "a = "+a+" b = "+b);
    swap(a,b);
    System.out.println("After Swap "+ "a = "+a+" b = "+b);

}
Run | Debug | Run main | Debug main
```

Java is strictly pass By Value

```
Invoking getSum(4,6): Result = 10
Invoking Swap(10,20)
Before Swap a = 10 b = 20
After Swap a = 10 b = 20
(base) kveeresh@Ks-MacBook-Air-2 Computer
```

↑ swapping

Output

```java
    public static void swap1(int[] arr){
        // int temp = arr[0];
        // arr[0] = arr[1];
        // arr[2] = temp;
        arr[0]=arr[0]+arr[1]-(arr[1]=arr[0]);
    }

    public static void invoke_swap(){
            System.out.println(x:"Invoking Swap(10,20)");

            int[] swapArr=new int[2];
            swapArr[0]=10;
            swapArr[1]=20;
            System.out.println("Before Swap "+ "a = "+swapArr[0]+" b = "+swapArr[1]);
            swap1(swapArr);
            System.out.println("After Swap "+ "a = "+swapArr[0]+" b = "+swapArr[1]);

    }
Run | Debug | Run main | Debug main
```

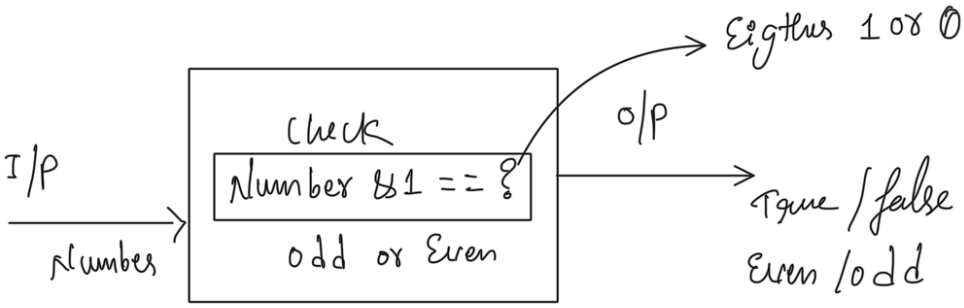
Equation

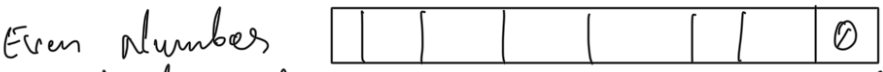
```
Invoking Swap(10,20)
Before Swap a = 10 b = 20
After Swap a = 20 b = 10
(base) kveeresh@Ks-MacBook-Air-2 Co
```

---

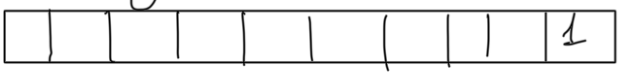
③ Write a function to check the Numbers is Even or odd

Binary decision
yes No
True false
1 0





For a Even Numbers last significant Bit is always 0



For a odd Numbers last significant Bit is always 1

We can also use modulas operator

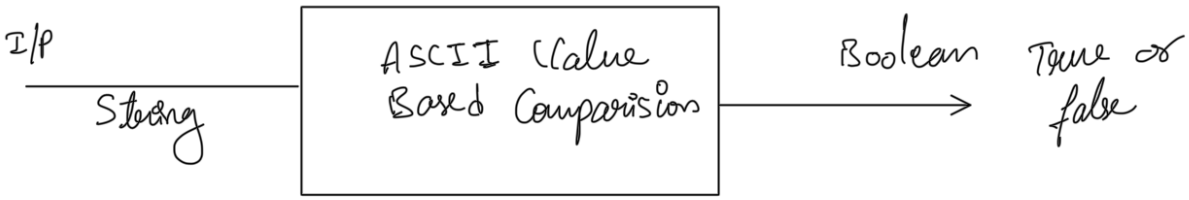
x % 2 == 0 ⟶ Even
x % 2 > 0 ⟶ odd

```java
    public static void isEven(int n){
        System.out.println("Entered Number "+n+" is: "+((n&1)==0?"Even":"Odd"));
    }
    public static void invoke_isEven(){
        System.out.println(x:"Invoking isEven(11)");
        isEven(n:11);
    }
Run | Debug | Run main | Debug main
    public static void main(String[] args) {
        invoke_getSum();
        invoke_swap();
        invoke_isEven();
    }
```

output

```
X:+ShowCodeDetailsInExceptionMess
rage/2ecbf2b3509965b320b049bd529e
Invoking getSum(4,6): Result =  1
Invoking Swap(10,20)
Before Swap a = 10 b = 20
After Swap a = 20 b = 10
Invoking isEven(11)
Entered Number 11 is: Odd
(base) kveeresh@Ks-MacBook-Air-2
```

④ Write a function to check given input is digit



String "146"

1	4	6

if Anything other than 0 & 9 return false

str[index] >='0'
&
str[index] <='9'

'0' ASCII Value of 0

```
43      public static boolean isDigit(String s){
44          for(int i=0; i<s.length();i++){
45              if(s.charAt(i)<'0'||s.charAt(i)>'9')return false;
46          }
47          return true;
48      }
49      public static void invoke_isDigit(){
50          System.out.println("Invoking is Digit for ''123'': "+ isDigit(s:"123"));
51          System.out.println("Invoking is Digit for ''12bA'': "+ isDigit(s:"12bA"));
52      }
        Run | Debug | Run main | Debug main
```

```
Entered Number 11 is: Odd
Invoking is Digit for ''123'': true
Invoking is Digit for ''12bA'': false
(base) kveeresh@Ks-MacBook-Air-2 Computer-Science %
Indexing completed.   Java: Ready
```

Input Reading

⑤ Function to accept Name as Input and printing simple Greeting.

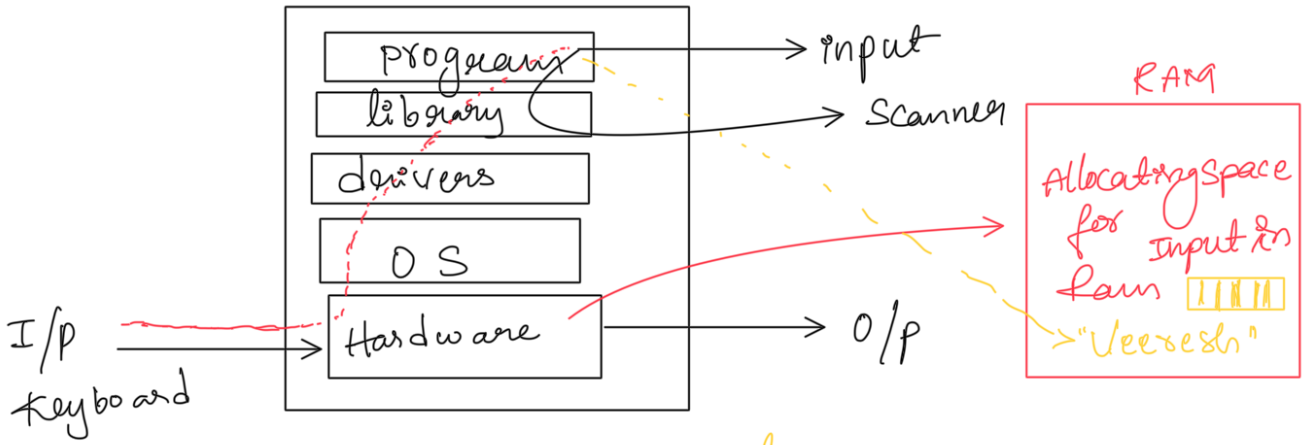


fig : Input Handling Input

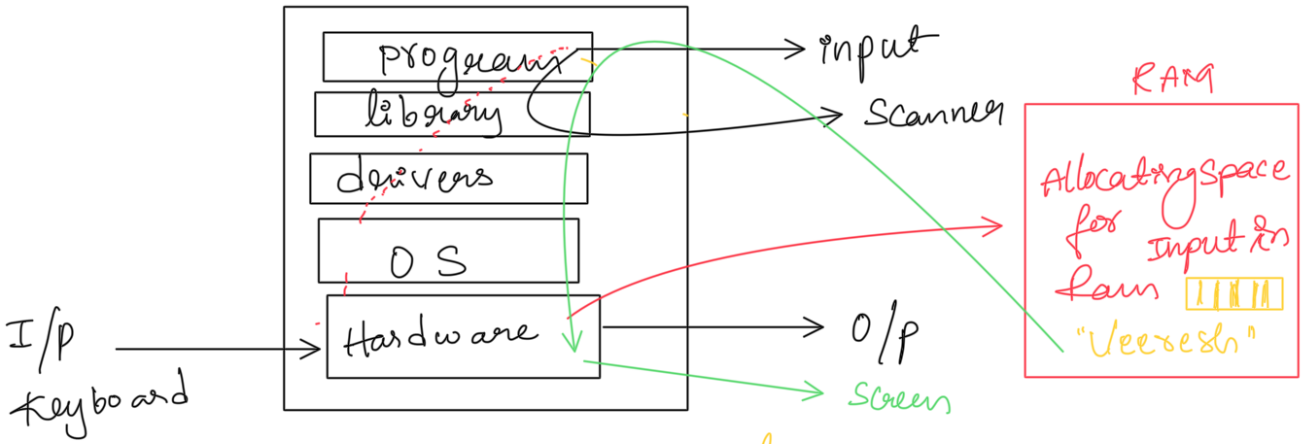
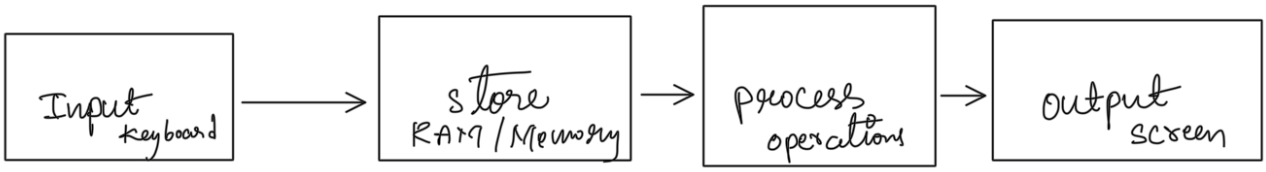


fig : Input Handling output



NOTE : From keyboard Input is taken as string

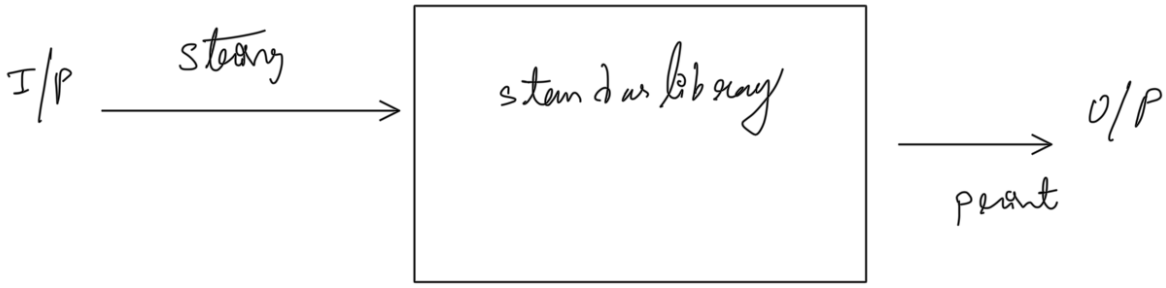
string => "20" ⟹ Integer 20

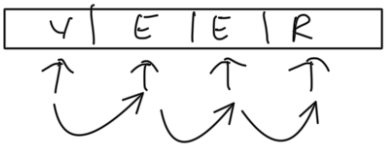
```java
    public static void greet(){
        System.out.println(x:"Eneter Name For Greeting");
        Scanner input = new Scanner(System.in);
        String s = input.nextLine();
        System.out.println("namaste "+ s);
        input.close();
    }

```

---

(6) Printing the ASCII Value of given String Input





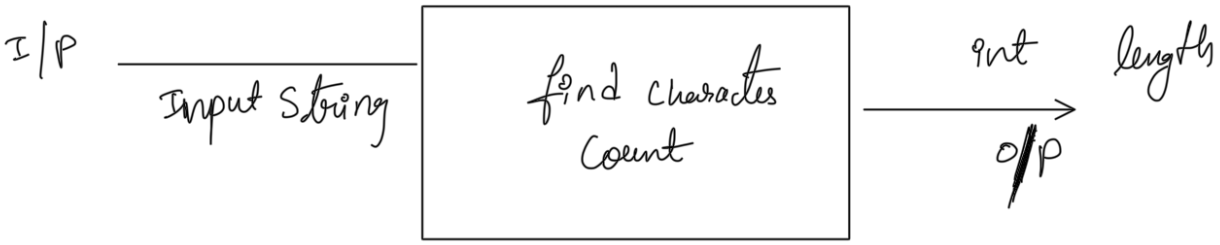

```java
    public static void printAscii(String s){
        for(int i=0;i<s.length();i++){
            char ch = s.charAt(i);
            System.out.println( ch +" Ascii = "+ (int) ch );
        }
    }
    public static void invoke_printAscii(){
        printAscii(s:"Hello Veeresh");
    }
    Run | Debug | Run main | Debug main
    public static void main(String[] args) {
```

PROBLEMS 18 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
Invoking Swap(10,20)
Before Swap a = 10 b = 20
After Swap a = 20 b = 10
Invoking isEven(11)
Entered Number 11 is: Odd
Invoking is Digit for ''123'': true
Invoking is Digit for ''12bA'': false
Eneter Name For Greeting
Linked in
namaste Linked in
H Ascii = 72
e Ascii = 101
l Ascii = 108
l Ascii = 108
o Ascii = 111
  Ascii = 32
V Ascii = 86
e Ascii = 101
e Ascii = 101
r Ascii = 114
e Ascii = 101
s Ascii = 115
h Ascii = 104
```

→ output

(7) Write a program to find string length



(1) no ores Each characters
(2) Counter
(3) traversing ores String

```java
72      public static int getStringLen(String s){
73          int c = 0;
74          for(char ch : s.toCharArray()){
75              c++;
76          }
77          return c;
78      }
        Run | Debug | Run main | Debug main
```

```java
72      public static int getStringLen(String s){
73          int c = 0;
74          // for(char ch : s.toCharArray()){
75          //     c++;
76          // }
77          return s.toCharArray().length;
78      }
        Run | Debug | Run main | Debug main
```

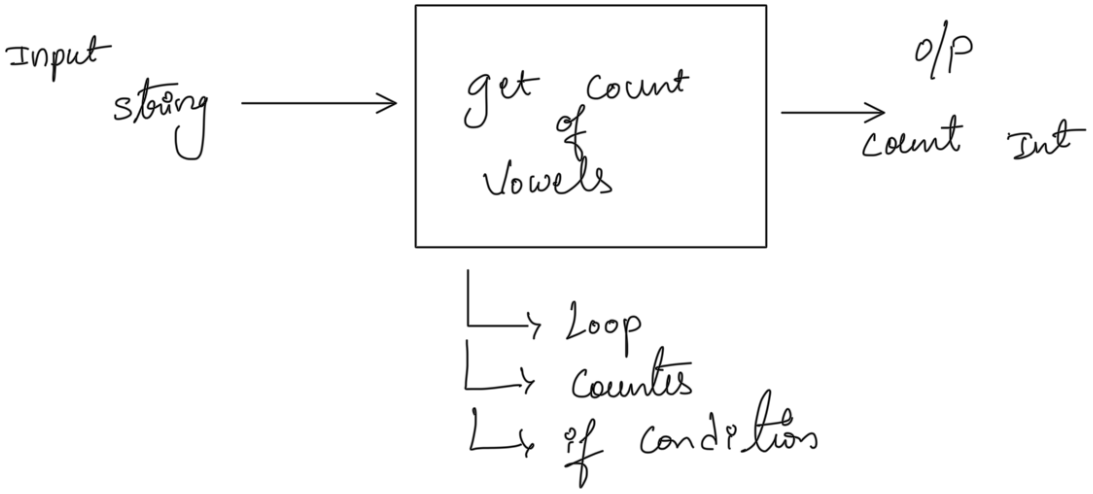
---

(8) Methods to get count of vowels in a given string

8 -> Count of Vowels

a	e	i	o	u

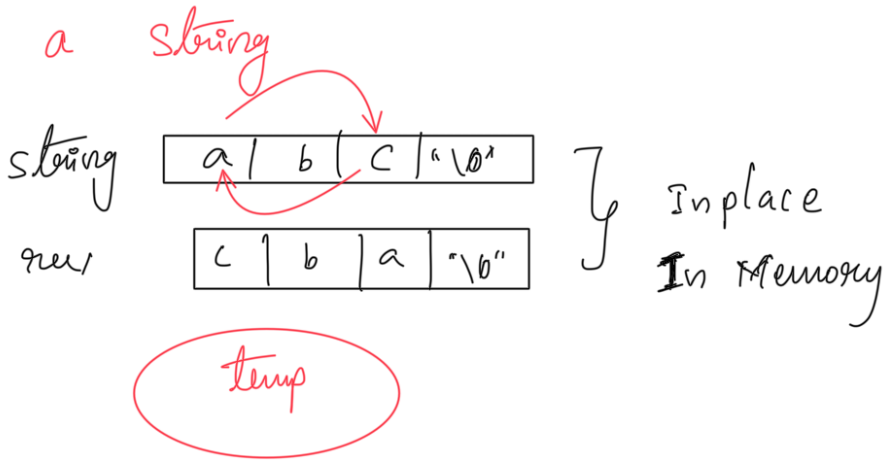
A	E	I	O	U




```java
79
80      public static int getVowelsCount(String s){
81          int c =0;
82          for(int i=0;i<s.length();i++){
83              char ch = s.charAt(i);
84              if (ch == 'a' || ch == 'i' || ch == 'e' || ch == 'o' || ch == 'u' ||
85              ch == 'A' || ch == 'E' || ch == 'I' || ch == 'O' || ch == 'U'  ) {
86                  c++;
87              }
88          }
89          return c;
90      }
        Run | Debug | Run main | Debug main
```

---

9 Reverse a String



In JAVA String is thread Safe & its immutable So we can use StringBuilder or characters Array, we can also use recursion, also we can use Stack.

```java
static String reveseString(String input){
StringBuilder output = new StringBuilder();

for (int index = input.length()-1 ; index >= 0; index--)
{
    output.append(input.charAt(index));
}

return output.toString();
}
```

```java
static String reveseStringV2(String input) {
    char[] inputArray = input.toCharArray();

    int leftIndex = 0;
    int rightIndex = input.length()-1;

    while(leftIndex < rightIndex)
    {
        char temp = inputArray[leftIndex];
        inputArray[leftIndex] = inputArray[rightIndex];
        inputArray[rightIndex] = temp;

        leftIndex++;
        rightIndex--;
    }

    String output = new String(inputArray);
    System.out.println(output);

    return output;
}

static void invoke_reverseString()
{
    String input1 = "Linked In";
    String output1 = reveseStringV2(input1);

    System.out.println("Input " + input1 + " Output " + output1);
}
```