Syntax for getting input from users:

Common functions to get input

Method	Description
`next()`	to get string without space from the user
`nextLine()`	to get string with spaces from the user
`nextInt()`	to get integer value from the user
`nextFloat()`	to get float value from the user
`nextByte()`	to get byte value from the user

Programs:

Swap of numbers with using third variable and without using third variable

```
class SwapTwoNumbers
{
public static void main(String []s)
      {
            int a,b;
            //Scanner class to read value
            Scanner sc=new Scanner(System.in);

            System.out.print("Enter value of a: ");
            a=sc.nextInt();
            System.out.print("Enter value of a: ");
            b=sc.nextInt();

            System.out.println("Before swapping - a: "+ a
+", b: " + b);
            ////using thrid variable
            int temp;
            temp=a;
            a=b;
            b=temp;
            //////////////////////
            System.out.println("After swapping  - a: "+ a
+", b: " + b);
      }
}
```

Swap2 numbers without using third variable

```
class SwapTwoNumbers
{
      public static void main(String []s)
      {
            int a,b;
            //Scanner class to read value
            Scanner sc=new Scanner(System.in);

            System.out.print("Enter value of a: ");
            a=sc.nextInt();
            System.out.print("Enter value of a: ");
            b=sc.nextInt();

            System.out.println("Before swapping - a: "+ a
+", b: " + b);
            ////without using thrid variable
            a=a+b;
            b=a-b;
            a=a-b;
            //////////////////////
            System.out.println("After swapping  - a: "+ a
+", b: " + b);
      }
}
```

Largest of 3 numbers

```
public class LargestNumber{

     public static void main(String []args)
     {
            int a=0,b=0,c=0;
            int largest=0;
            //Scanner class to take user input.
            Scanner X = new Scanner(System.in);

            System.out.print("Enter First No. :");
            a = X.nextInt(); //read integer number

            System.out.print("Enter Second No. :");
            b = X.nextInt(); //read integer number

            System.out.print("Enter Third No. :");
            c = X.nextInt(); //read integer number

            if( a>b && a> c)
                largest = a;
            else if(b>a && b>c)
                largest = b;
            else
                largest = c;

            System.out.println("Lagest Number is :
"+largest);
     }
}
```

Simple interest

```
public class SimpleInterest{

    public static void main(String []args)
    {
        double p=0,r=0,t=0,si=0;

        //Scanner class to take user input.
        Scanner X = new Scanner(System.in);

        System.out.print("Enter Principle : ");
        p = X.nextFloat();

        System.out.print("Enter Rate of Interest: ");
        r =  X.nextFloat();

        System.out.print("Enter Time in years : ");
        t =  X.nextFloat();


        //Formula of simple interest.
        si = (p*r*t)/100;

        System.out.print("Simple Interest is :"+si+"\n");

    }
}
```

Factorial

Factorial Program in Java: Factorial of n is the *product of all positive descending integers*. Factorial of *n* is denoted by n!. For example:

1. 4! = 4*3*2*1 = 24
2. 5! = 5*4*3*2*1 = 120

```
public class Factorial
{
    public static void main(String args[]){
        int num;
        long factorial;

        Scanner bf=new Scanner(System.in);

        //input an integer number
        System.out.print("Enter any integer number: ");
        num= bf.nextInt();

        //find factorial
        factorial=1;
        for(int loop=num; loop>=1; loop--)
            factorial*=loop;

        System.out.println("Factorial of " + num + " is: " + factorial);
    }
}
```

Palindrome:

A Palindrome Number is a number that even when reversed is same as original number

Examples of Palindrome Number

```
121, 393, 34043, 111, 555, 48084
```

```
Class Palindrome
{
public static void main(String args[]){

        int num,tNum,sum=0,r;

        Scanner bf=new Scanner(System.in);

        //input an integer number
        System.out.print("Enter any integer number: ");
        num= bf.nextInt();

        //find reverse number
        tNum=num;

        while(tNum>0)
        {
            r=tNum%10;
            tNum=tNum/10;
            sum = (sum*10) + r;

        }

        //check inputted number with reversed number
        if(num==sum)
            System.out.println(num + " is a Palindrome
Number.");
        else
            System.out.println(num + " is not a Palindrome
Number.");
    }
}
```

Here are the execution steps that takes place:

Palindrome execution steps			
num	num != 0	remainder	reversedInteger
121	true	1	0 * 10 + 1 = 1
12	true	2	1 * 10 + 2 = 12
1	true	1	12 * 10 + 1 = 121
0	false	-	121

Reverse of number:

```
public class ReverseNumber{

     public static void main(String []args){
        int number,r;

        Scanner sc=new Scanner(System.in);

        //Read Number
        System.out.print("Enter an integer number: ");
        number=sc.nextInt();

        //calculate reverse number
        int reverse_number=0;
        while(number>0)
        {
            r=number%10;
            reverse_number = (reverse_number*10) +r;
            number/=10;
        }

        System.out.println("Reverse Number is: " + reverse_number);

     }
}
```

Reverse of string

The **java string length()** method length of the string. It returns count of total number of characters. The length of java string is same as the unicode code units of the string.

Ex:String S1=”java”;

S1.length()=4

The **java**.lang.String.**charAt**() **method** returns the char value at the specified index. An index ranges from 0 to length() - 1. The first char value of the sequence is at index 0, the next at index 1, and so on, as for array

S1.charAt(1)→will give output as ‘a’because indexing start from 0

```
public class ReverseString
{
    public static void main(String args[]){
        String str;
        String rStr;
        Scanner bf=new Scanner(System.in);

        //input an integer number
        System.out.print("Enter any string: ");
        str=bf.nextLine();

        //Reversing String
        rStr="";
        for(int loop=str.length()-1; loop>=0; loop--)
            rStr= rStr + str.charAt(loop);

        System.out.println("Reversed string is: " + rStr);
    }
}
```

Extract number from string

Use `String.replaceAll(String regex, String replacement)` to replace all occurrences of a substring (matching argument `regex`) with `replacement` string

Regrex:Regular expression

```
class ExtractNumberFromString
{
  public static void main(String[] args)
  {
      String str;
      String numbers;

      Scanner SC=new Scanner(System.in);

      System.out.print("Enter string that contains numbers: 
");
      str=SC.nextLine();

      //extracting string
      numbers=str.replaceAll("[^0-9]", "");

      System.out.println("Numbers are: " + numbers);
  }
}
```

Count number of words in sentence

```
class CountWords
{

    public static void main(String args[])
    {
        String text;
        int countWords=1;

        Scanner SC=new Scanner(System.in);

        System.out.print("Enter string: ");
        text=SC.nextLine();

        //word count
        for(int i=0; i<text.length()-1; i++)
        {
            if(text.charAt(i)==' ' && text.charAt(i+1)!='
')
                countWords++;
        }

        System.out.println("Total number of words in string
are: "+ countWords);
        //since last word does not contain and character
after that

    }
}
```

Multiplication table:

```
public class PrintTable {

    public static void main(String args[]){

        int number;
        Scanner SC=new Scanner(System.in);

        //input integer number
        System.out.print("Enter an integer postive number: ");
        number=SC.nextInt();

        //print table
        System.out.println("Table of " + number + " is ");
        for(int loop=1; loop<=10; loop++){
            System.out.println(number*loop);
        }
    }
}
```

armstromg number

Armstrong Number in Java: A positive number is called **armstrong number** if it is equal to the sum of cubes of its digits for example 0, 1, 153, 370, 371, 407 etc.

Let's try to understand why **153** is an Armstrong number.

1. 153 = (1*1*1)+(5*5*5)+(3*3*3)
2. where:
3. (1*1*1)=1
4. (5*5*5)=125
5. (3*3*3)=27
6. So:
7. 1+125+27=153

```
 class ArmstrongExample
 {
   public static void main(String[] args)
   {
    int c=0,a,temp;
    int n=153;//It is the number to check armstrong
     temp=n;
     while(n>0)
     {
     a=n%10;
     n=n/10;
     c=c+(a*a*a);
     }
     if(temp==c)
     System.out.println("armstrong number");
     else
         System.out.println("Not armstrong number");
    }
}
```

Perfect Number:

Any number can be a Java Perfect Number, If the sum of its positive divisors excluding the number itself is equal to that number. For example, 28 is a perfect number because 28 is divisible by 1, 2, 4, 7, 14 and 28 and the sum of these values are: 1 + 2 + 4 + 7 + 14 = 28 (Remember, we have to exclude the number itself. That’s why we haven’t added 28 here). Some of the perfect numbers are 6, 28, 496, 8128 and 33550336 so on

```
public class PerfectNumberUsingFor {


        public static void main(String[] args)
             {
               int i, Number, Sum = 0 ;
               scanner sc = new Scanner(System.in);
               System.out.println("\n Please Enter any Number: ");
               Number = sc.nextInt();

               for(i = 1 ; i < Number ; i++)
              {
                       if(Number % i == 0)
                      {
                               Sum = Sum + i;
                       }
               }
               if (Sum == Number) {
                       System.out.format("\n% d is a Perfect Number", Number);
               }
               else {
                       System.out.format("\n%  d  is  NOT  a  Perfect  Number",
Number);
               }
        }
}
```

Fibnocci series:

The Fibonacci series is a series where the next term is the sum of pervious two terms. The first two terms of the Fibonacci sequence is 0 followed by 1.

```
The Fibonacci sequence: 0, 1, 1, 2, 3, 5, 8, 13, 21, ...
```

```
public class Fibonacci
{
    public static void main(String[] args)
 {
        int n = 10, t1 = 0, t2 = 1;
        System.out.print("First " + n + " terms: ");

        for (int i = 1; i <= n; ++i)
        {
            System.out.print(t1 + " + ");

            int sum = t1 + t2;
            t1 = t2;
            t2 = sum;
        }
    }
}
```

Comparing two strings

First method:

This method compares this string to the specified object. The result is true if and only if the argument is not null and is a String object that represents the same sequence of characters as this object.

```
public class StringCompareequl

{

   public static void main(String []args){
      String s1 = "tutorial";
      String s2 = "tutorial";
      String s3 = new String ("Tutorials");
      System.out.println(s1.equals(s2));
      System.out.println(s2.equals(s3));
   }
}
```

The above code sample will produce the following result.

```
true
false
```

Second method:

String compare by == operator

```
public class StringCompareequl{
   public static void main(String []args){
      String s1 = "tutorial";
      String s2 = "tutorial";
      String s3 = new String ("Tutorials");
      System.out.println(s1 == s2);
      System.out.println(s2 == s3);
   }
}
```

The above code sample will produce the following result.

```
true
false
```

Searching a word in string

```java
public class HelloWorld {
   public static void main(String[] args) {
      String text = "The cat is on the table";
      System.out.print(text.contains("the"));
   }
}
```

Result

The above code sample will produce the following result.

```
true
```

Conerting string to upper case

```java
public class StringToUpperCaseEmp {
   public static void main(String[] args) {
      String str = "string abc touppercase ";
      String strUpper = str.toUpperCase();
      System.out.println("Original String: " + str);
      System.out.println("String changed to upper case: " + strUpper);
   }
}
```

Result

The above code sample will produce the following result.

```
Original String: string abc touppercase
String changed to upper case: STRING ABC TOUPPERCASE
```

Concatination of string:

```
public class HelloWorld {

   public static void main(String []args) {

      String s = "Hello";

      s = s.concat("word");

      System.out.print(s);

   }

}
```

Result

The above code sample will produce the following result. The result may vary.

```
Helloword
```

string split

```
public class HelloWorld {

   public static void main(String args[]) {

      String s1 = "t u t o r i a l s";

      String[] words = s1.split("\\s");

      for(String w:words) {

         System.out.println(w);

      }

   }

}
```

Result

The above code sample will produce the following result.

```
t
u
t
o
r
```

```
i
a
l
s
```

Convert String to integer

```
 class Demo
{
public static void main(String args[]){
//Declaring String variable
String s="200";
//Converting String into int using Integer.parseInt()
int i=Integer.parseInt(s);
//Printing value of i
System.out.println(i);
}}
```

Duplicate in string:

```
public class DuplicateSample {
   public static void main(String args[]){
      String str = "malayalam";
      char[] myArray = str.toCharArray();



      for(int i=0; i<myArray.length-1; i++){

         for (int j=i+1; j<myArray.length; j++){

            if(myArray[i] == myArray[j]){

               myArray = ArrayUtils.remove(myArray, j);

            }

         }

      }

      System.out.println("String value after deleting the duplicate values
:"+Arrays.toString(myArray));

   }

}
```

Output

```
String value after deleting the duplicate values :[m, a, l, y]
```