**Java program to reverse a number using for**

### Program 1: Reverse a number using while Loop

The program will prompt user to input the number and then it will reverse the same number using while loop.

import java.util.Scanner;

class ReverseNumberWhile

{

public static void main(String args[])

{

int num=0;

int reversenum =0;

System.out.println("Input your number and press enter: ");

//This statement will capture the user input

Scanner in = new Scanner(System.in);

//Captured input would be stored in number num

num = in.nextInt();

//While Loop: Logic to find out the reverse number

while( num != 0 )

{

reversenum = reversenum \* 10;

reversenum = reversenum + num%10;

num = num/10;

}

System.out.println("Reverse of input number is: "+reversenum);

}

}

### Program 2: Reverse a number using for Loop

import java.util.Scanner;

class ForLoopReverseDemo

{

public static void main(String args[])

{

int num=0;

int reversenum =0;

System.out.println("Input your number and press enter: ");

//This statement will capture the user input

Scanner in = new Scanner(System.in);

//Captured input would be stored in number num

num = in.nextInt();

/\* for loop: No initialization part as num is already

\* initialized and no increment/decrement part as logic

\* num = num/10 already decrements the value of num

\*/

for( ;num != 0; )

{

reversenum = reversenum \* 10;

reversenum = reversenum + num%10;

num = num/10;

}

System.out.println("Reverse of specified number is: "+reversenum);

}

}

### Program 3: Reverse a number using recursion

import java.util.Scanner;

class RecursionReverseDemo

{

//A method for reverse

public static void reverseMethod(int number) {

if (number < 10) {

System.out.println(number);

return;

}

else {

System.out.print(number % 10);

//Method is calling itself: recursion

reverseMethod(number/10);

}

}

public static void main(String args[])

{

int num=0;

System.out.println("Input your number and press enter: ");

Scanner in = new Scanner(System.in);

num = in.nextInt();

System.out.print("Reverse of the input number is:");

reverseMethod(num);

System.out.println();

}

}

**Example: Reverse an already initialized number**  
In all the above programs we are prompting user for the input number, however if do not want the user interaction part and want to reverse an initialized number then this is how you can do it.

class ReverseNumberDemo

{

public static void main(String args[])

{

int num=123456789;

int reversenum =0;

while( num != 0 )

{

reversenum = reversenum \* 10;

reversenum = reversenum + num%10;

num = num/10;

}

System.out.println("Reverse of specified number is: "+reversenum);

}

}

**Check whether String is palindrome**

**Program 1: Palindrome check Using Stack**

import java.util.Stack;

import java.util.Scanner;

class PalindromeTest {

public static void main(String[] args) {

System.out.print("Enter any string:");

Scanner in=new Scanner(System.in);

String inputString = in.nextLine();

Stack stack = new Stack();

for (int i = 0; i < inputString.length(); i++) {

stack.push(inputString.charAt(i));

}

String reverseString = "";

while (!stack.isEmpty()) {

reverseString = reverseString+stack.pop();

}

if (inputString.equals(reverseString))

System.out.println("The input String is a palindrome.");

else

System.out.println("The input String is not a palindrome.");

}

}

**Program 2: Palindrome check Using Queue**

import java.util.Queue;

import java.util.Scanner;

import java.util.LinkedList;

class PalindromeTest {

public static void main(String[] args) {

System.out.print("Enter any string:");

Scanner in=new Scanner(System.in);

String inputString = in.nextLine();

Queue queue = new LinkedList();

for (int i = inputString.length()-1; i >=0; i--) {

queue.add(inputString.charAt(i));

}

String reverseString = "";

while (!queue.isEmpty()) {

reverseString = reverseString+queue.remove();

}

if (inputString.equals(reverseString))

System.out.println("The input String is a palindrome.");

else

System.out.println("The input String is not a palindrome.");

}

}

**Program 3: Using for loop/While loop and String function charAt**

import java.util.Scanner;

class PalindromeTest {

public static void main(String args[])

{

String reverseString="";

Scanner scanner = new Scanner(System.in);

System.out.println("Enter a string to check if it is a palindrome:");

String inputString = scanner.nextLine();

int length = inputString.length();

for ( int i = length - 1 ; i >= 0 ; i-- )

reverseString = reverseString + inputString.charAt(i);

if (inputString.equals(reverseString))

System.out.println("Input string is a palindrome.");

else

System.out.println("Input string is not a palindrome.");

}

}

**Program 4: using recursion:**

package beginnersbook.com;

import java.util.Scanner;

class PalindromeCheck

{

//My Method to check

public static boolean isPal(String s)

{ // if length is 0 or 1 then String is palindrome

if(s.length() == 0 || s.length() == 1)

return true;

if(s.charAt(0) == s.charAt(s.length()-1))

/\* check for first and last char of String:

\* if they are same then do the same thing for a substring

\* with first and last char removed. and carry on this

\* until you string completes or condition fails

\* Function calling itself: Recursion

\*/

return isPal(s.substring(1, s.length()-1));

/\* If program control reaches to this statement it means

\* the String is not palindrome hence return false.

\*/

return false;

}

public static void main(String[]args)

{

//For capturing user input

Scanner scanner = new Scanner(System.in);

System.out.println("Enter the String for check:");

String string = scanner.nextLine();

/\* If function returns true then the string is

\* palindrome else not

\*/

if(isPal(string))

System.out.println(string + " is a palindrome");

else

System.out.println(string + " is not a palindrome");

}

}

# Java program for linear search

This program uses [linear search algorithm](http://en.wikipedia.org/wiki/Linear_search) to find out a number among all other numbers entered by user.

/\* Program: Linear Search Example

\* Written by: Chaitanya from beginnersbook.com

\* Input: Number of elements, element's values, value to be searched

\* Output:Position of the number input by user among other numbers\*/

import java.util.Scanner;

class LinearSearchExample

{

public static void main(String args[])

{

int counter, num, item, array[];

//To capture user input

Scanner input = new Scanner(System.in);

System.out.println("Enter number of elements:");

num = input.nextInt();

//Creating array to store the all the numbers

array = new int[num];

System.out.println("Enter " + num + " integers");

//Loop to store each numbers in array

for (counter = 0; counter < num; counter++)

array[counter] = input.nextInt();

System.out.println("Enter the search value:");

item = input.nextInt();

for (counter = 0; counter < num; counter++)

{

if (array[counter] == item)

{

System.out.println(item+" is present at location "+(counter+1));

/\*Item is found so to stop the search and to come out of the

\* loop use break statement.\*/

break;

}

}

if (counter == num)

System.out.println(item + " doesn't exist in array.");

}

}

# Java program to perform binary search –

#### Example Program to perform binary search on a list of integer numbers

This program uses [binary search algorithm](http://en.wikipedia.org/wiki/Binary_search_algorithm) to search an element in given list of elements.

/\* Program: Binary Search Example

\* Written by: Chaitanya from beginnersbook.com

\* Input: Number of elements, element's values, value to be searched

\* Output:Position of the number input by user among other numbers\*/

import java.util.Scanner;

class BinarySearchExample

{

public static void main(String args[])

{

int counter, num, item, array[], first, last, middle;

//To capture user input

Scanner input = new Scanner(System.in);

System.out.println("Enter number of elements:");

num = input.nextInt();

//Creating array to store the all the numbers

array = new int[num];

System.out.println("Enter " + num + " integers");

//Loop to store each numbers in array

for (counter = 0; counter < num; counter++)

array[counter] = input.nextInt();

System.out.println("Enter the search value:");

item = input.nextInt();

first = 0;

last = num - 1;

middle = (first + last)/2;

while( first <= last )

{

if ( array[middle] < item )

first = middle + 1;

else if ( array[middle] == item )

{

System.out.println(item + " found at location " + (middle + 1) + ".");

break;

}

else

{

last = middle - 1;

}

middle = (first + last)/2;

}

if ( first > last )

System.out.println(item + " is not found.\n");

}

}

# Java program to print Floyd’s triangle –

This program will prompt user for number of rows and based on the input, it would print the [Floyd’s triangle](http://en.wikipedia.org/wiki/Floyd%27s_triangle) having the same number of rows.

/\* Program: It Prints Floyd's triangle based on user inputs

\* Written by: Chaitanya from beginnersbook.com

\* Input: Number of rows

\* output: floyd's triangle\*/

import java.util.Scanner;

class FloydTriangleExample

{

public static void main(String args[])

{

int rows, number = 1, counter, j;

//To get the user's input

Scanner input = new Scanner(System.in);

System.out.println("Enter the number of rows for floyd's triangle:");

//Copying user input into an integer variable named rows

rows = input.nextInt();

System.out.println("Floyd's triangle");

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

for ( counter = 1 ; counter <= rows ; counter++ )

{

for ( j = 1 ; j <= counter ; j++ )

{

System.out.print(number+" ");

//Incrementing the number value

number++;

}

//For new line

System.out.println();

}

}

}

**Java program to sum the elements of an array**

In this tutorial we will see **how to sum up all the elements of an array**.

Program 1: No user interaction

class SumOfArray{

public static void main(String args[]){

int[] array = {10, 20, 30, 40, 50, 10};

int sum = 0;

//Advanced for loop

for( int num : array) {

sum = sum+num;

}

System.out.println("Sum of array elements is:"+sum);

}

}

Program 2: User enters the array’s elements

/\*\*

\* @author: BeginnersBook.com

\* @description: User would enter the 10 elements

\* and the program will store them into an array and

\* will display the sum of them.

\*/

import java.util.Scanner;

class SumDemo{

public static void main(String args[]){

Scanner scanner = new Scanner(System.in);

int[] array = new int[10];

int sum = 0;

System.out.println("Enter the elements:");

for (int i=0; i<10; i++)

{

array[i] = scanner.nextInt();

}

for( int num : array) {

sum = sum+num;

}

System.out.println("Sum of array elements is:"+sum);

}

}

# Java Program to calculate area and circumference of circle

import java.util.Scanner;

class CircleDemo

{

static Scanner sc = new Scanner(System.in);

public static void main(String args[])

{

System.out.print("Enter the radius: ");

/\*We are storing the entered radius in double

\* because a user can enter radius in decimals

\*/

double radius = sc.nextDouble();

//Area = PI\*radius\*radius

double area = Math.PI \* (radius \* radius);

System.out.println("The area of circle is: " + area);

//Circumference = 2\*PI\*radius

double circumference= Math.PI \* 2\*radius;

System.out.println( "The circumference of the circle is:"+circumference) ;

}

}

**Write programs to find out the factorial of a number using recursion**.

**Program 1:**  
Program will prompt user for the input number. Once user provide the input, the program will calculate the factorial for the provided input number.

/\*\*

\* @author: BeginnersBook.com

\* @description: User would enter the 10 elements

\* and the program will store them into an array and

\* will display the sum of them.

\*/

import java.util.Scanner;

class FactorialDemo{

public static void main(String args[]){

//Scanner object for capturing the user input

Scanner scanner = new Scanner(System.in);

System.out.println("Enter the number:");

//Stored the entered value in variable

int num = scanner.nextInt();

//Called the user defined function fact

int factorial = fact(num);

System.out.println("Factorial of entered number is: "+factorial);

}

static int fact(int n)

{

int output;

if(n==1){

return 1;

}

//Recursion: Function calling itself!!

output = fact(n-1)\* n;

return output;

}

}

**Program 2:**   
If you do not want user intervention and simply want to specify the number in program itself then refer this example.

class FactorialDemo2{

public static void main(String args[]){

int factorial = fact(4);

System.out.println("Factorial of 4 is: "+factorial);

}

static int fact(int n)

{

int output;

if(n==1){

return 1;

}

//Recursion: Function calling itself!!

output = fact(n-1)\* n;

return output;

}

}

**Java Program to check Even or Odd nums**

import java.util.Scanner;

class CheckEvenOdd

{

public static void main(String args[])

{

int num;

System.out.println("Enter an Integer number:");

//The input provided by user is stored in num

Scanner input = new Scanner(System.in);

num = input.nextInt();

/\* If number is divisible by 2 then it's an even number

\* else odd number\*/

if ( num % 2 == 0 )

System.out.println("Entered number is even");

else

System.out.println("Entered number is odd");

}

}

### Method 1: Binary to Decimal conversion using Integer.parseInt() method

import java.util.Scanner;

class BinaryToDecimal {

public static void main(String args[]){

Scanner input = new Scanner( System.in );

System.out.print("Enter a binary number: ");

String binaryString =input.nextLine();

System.out.println("Output: "+Integer.parseInt(binaryString,2));

}

}

### Method 2: Conversion without using parseInt

public class Details {

public int BinaryToDecimal(int binaryNumber){

int decimal = 0;

int p = 0;

while(true){

if(binaryNumber == 0){

break;

} else {

int temp = binaryNumber%10;

decimal += temp\*Math.pow(2, p);

binaryNumber = binaryNumber/10;

p++;

} } return decimal;

}

public static void main(String args[]){

Details obj = new Details();

System.out.println("110 --> "+obj.BinaryToDecimal(110));

System.out.println("1101 --> "+obj.BinaryToDecimal(1101));

System.out.println("100 --> "+obj.BinaryToDecimal(100));

System.out.println("110111 --> "+obj.BinaryToDecimal(110111));

} }

**Java Program to find duplicate Characters in a String**

This program would find out the duplicate characters in a String and would display the count of them.

import java.util.HashMap;

import java.util.Map;

import java.util.Set;

public class Details {

public void countDupChars(String str){

//Create a HashMap

Map<Character, Integer> map = new HashMap<Character, Integer>();

//Convert the String to char array

char[] chars = str.toCharArray();

/\* logic: char are inserted as keys and their count

\* as values. If map contains the char already then

\* increase the value by 1

\*/

for(Character ch:chars){

if(map.containsKey(ch)){

map.put(ch, map.get(ch)+1);

} else {

map.put(ch, 1);

} }

//Obtaining set of keys

Set<Character> keys = map.keySet();

/\* Display count of chars if it is

\* greater than 1. All duplicate chars would be

\* having value greater than 1.

\*/

for(Character ch:keys){

if(map.get(ch) > 1){

System.out.println("Char "+ch+" "+map.get(ch));

} } }

public static void main(String a[]){

Details obj = new Details();

System.out.println("String: BeginnersBook.com");

System.out.println("-------------------------");

obj.countDupChars("BeginnersBook.com");

System.out.println("\nString: ChaitanyaSingh");

System.out.println("-------------------------");

obj.countDupChars("ChaitanyaSingh");

System.out.println("\nString: #@$@!#$%!!%@");

System.out.println("-------------------------");

obj.countDupChars("#@$@!#$%!!%@");

}

}

# Flow control in try-catch-finally blocks

### Flow of control in try/catch/finally blocks:

1. If exception occurs in try block’s body then control immediately transferred(**skipping rest of the statements in try block**) to the catch block. Once catch block finished execution then [finally block](http://beginnersbook.com/2013/04/java-finally-block/) and after that rest of the program.
2. If there is no exception occurred in the code which is present in try block then first, the try block gets executed completely and then control gets transferred to finally block (**skipping catch blocks**).
3. If a [return statement](http://beginnersbook.com/2013/05/java-finally-return/) is encountered either in try or catch block. In such case also **finally runs**. Control first goes to finally and then it returned back to **return statement.**
4. **Try block can be executed without catch block by adding finally block and throws exception keyword in method.**

**Consider the below example to understand above mentioned points:**

class TestExceptions {

static void myMethod(int testnum) throws Exception {

System.out.println ("start - myMethod");

if (testnum == 12)

throw new Exception();

System.out.println("end - myMethod");

return;

}

public static void main(String args[]) {

int testnum = 12;

try {

System.out.println("try - first statement");

myMethod(testnum);

System.out.println("try - last statement");

}

catch ( Exception ex) {

System.out.println("An Exception");

}

finally {

System. out. println( "finally") ;

}

System.out.println("Out of try/catch/finally - statement");

}

}

**Output:**

try - first statement

start - myMethod

An Exception

finally

Out of try/catch/finally - statement

# How to Catch multiple exceptions

A method can [throw more than one exceptions](http://beginnersbook.com/2013/04/throw-in-java/). However that method needs to declare all the [checked exceptions](http://beginnersbook.com/2013/04/java-checked-unchecked-exceptions-with-examples/) it can throw (optionally method can declare the super class of exception if it is common among all the exceptions).

public class MyExceptionClass

{

public void myMethod() throws ArithmeticException, ArrayIndexOutOfBoundsException

{

.....

.....

}

}

public class Demo

{

public void display()

{

MyExceptionClass obj= new MyExceptionClass();

try

{

//calling method of parent class

obj.myMethod();

}catch(ArithmeticException ex1)

{

System.out.println("Arithmetic Exception occurred!!");

}

catch(ArrayIndexOutOfBoundsException ex2)

{

System.out.println("Array Index Out of Bounds exception occurred!!");

}

}

}