**Sample Programs**

**String Palindrome:**

**public** **class** StringPalindrome {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

String str="madam";

StringBuffer sb1 = **new** StringBuffer(str);

StringBuffer sb2 = **new** StringBuffer(sb1);

sb1.reverse();

**if**(String.*valueOf*(sb1).compareTo(String.*valueOf*(sb2))==0)

{

System.***out***.println("Given String is Palindrome");

}

**else**

System.***out***.println("Given String is not a Palindrome");

}

}

Out Put: Given String is Palindrome

**Palindrome Number:**

**public** **class** PalindromeNumber {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

String str="12121";

String reverseString="";

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

{

reverseString=reverseString+str.charAt(i);

}

**if**(String.*valueOf*(str).compareTo(String.*valueOf*(reverseString))==0)

{

System.***out***.println("Give Number "+str+" is Palindrome");

}

**else**

System.***out***.println("Give Number "+str+" is not a Palindrome");

}

}

Out Put: Given String is Palindrome

Give Number 12121 is Palindrome

**Amstrong Number:**

**public** **class** AmstrongNumber {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**int** sum=0, temp;

**int** n=153;

temp=n;

**while**(temp!=0)

{

**int** r=temp%10;

sum=sum+r\*r\*r;

temp=temp/10;

}

**if**(n==sum)

{

System.***out***.println(n+" is Amstrong Number");

}

**else**

System.***out***.println(n+" is not an Amstrong Number");

}

}

Out Put: 153 is Amstrong Number

**Integer Sorting:**

**public** **class** IntegersSorting {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**int** a[] = **new** **int**[] {2,99,5,0,1,67,32,-99,-43,12,75};

**int** temp;

**for**(**int** i=0;i<a.length;i++)

{

**for**(**int** j=i+1;j<a.length;j++)

{

**if**(a[i]>a[j])

{

temp=a[i];

a[i]=a[j];

a[j]=temp;

}

}

}

System.***out***.println("Ascending Order");

**for**(**int** i=0;i<a.length;i++)

{

System.***out***.println(a[i]);

}

}

}

Out Put:

Ascending Order :

-99

-43

0

1

2

5

12

32

67

75

99

**Alphabets Sorting:**

**package** com.penchu.logics;

**public** **class** AlphabetSorting {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**char** a[] = **new** **char**[] {'D','H','A','F','G','B','E','C'};

**int** temp;

**for**(**int** i=0;i<a.length;i++)

{

**for**(**int** j=i+1;j<a.length;j++)

{

**if**(a[i]>a[j])

{

temp=a[i];

a[i]=a[j];

a[j]=(**char**)temp;

}

}

}

System.***out***.println("Ascending Order :");

**for**(**int** i=0;i<a.length;i++)

{

System.***out***.println(a[i]);

}

}

}

Out Put:

Ascending Order :

A

B

C

D

E

F

G

H

**String Words Reverse:**

**public** **class** StringWordsReverse {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

String s="welcome to Bangalore City";

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

String reverseString="";

**for**(**int** i=0;i<words.length;i++)

{

String word = words[i];

String reverseWord =" ";

**for**(**int** j=word.length()-1;j>=0;j--)

{

reverseWord = reverseWord+word.charAt(j);

}

reverseString=reverseString+reverseWord+"";

}

System.***out***.println(reverseString);

}

}

Out Put:

emoclew ot erolagnaB ytiC

**String Reverse:**

**public** **class** ReverseString {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

String str="welcome";

String reverseString="";

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

{

reverseString=reverseString+str.charAt(i);

}

System.***out***.println(reverseString);

}

}

Out Put:

Emoclew

**Swapping:**

**package** com.penchu.logics;

**public** **class** SwappingNumbers {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**int** num1=10;

**int** num2=20;

num1=num1+num2;

num2=num1-num2;

num1=num1-num2;

System.***out***.println(num1);

System.***out***.println(num2);

}

}

Out Put:

20

10

**Frequency Occurency of Charecter:**

**public** **class** FrequenceOccurence {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

String str="communication";

**int** i=str.length();

**for**(**char** c='a';c<='z';c++)

{

**int** k=0;

**for**(**int** j=0;j<i;j++)

{

**char** ch=str.charAt(j);

**if**(ch==c)

{

k++;

}

}

**if**(k>0)

{

System.***out***.println("The Charecter "+c+" has occured "+k+" times");

}

}

}

}

Out Put:

The Charecter a has occured 1 times

The Charecter c has occured 2 times

The Charecter i has occured 2 times

The Charecter m has occured 2 times

The Charecter n has occured 2 times

The Charecter o has occured 2 times

The Charecter t has occured 1 times

The Charecter u has occured 1 times

**Largest & Smallest Number:**

**public** **class** FindLargestSmallestNumber {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**int** numbers[] = **new** **int**[]{34,0,99,21,-99,78,-80};

**int** smallest = numbers[0];

**int** largest = numbers[0];

**for**(**int** i=1;i<numbers.length;i++)

{

**if**(numbers[i]>largest)

largest=numbers[i];

**else** **if**(numbers[i]<smallest)

smallest=numbers[i];

}

System.***out***.println("Largest Number is:"+largest);

System.***out***.println("Smallest Number is:"+smallest);

}

}

Out Put:

Largest Number is:99

Smallest Number is:-99

**Factorial Number:**

**public** **class** FactorialNumber {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**int** n=5;

**int** factorial=n;

**for**(**int** i=n-1;i>1;i--)

{

factorial=factorial\*i;

}

System.***out***.println("Factorial of Given Number is:"+factorial);

}

}

Out Put:

Factorial of Given Number is:120

**Prime Numbers:**

**public** **class** PrimeNumbers {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**int** count=0;

**for**(**int** i=1;i<=30;i++)

{

count=0;

**for**(**int** j=2;j<=i/2;j++)

{

**if**(i%j==0)

{

count++;

**break**;

}

}

**if**(count==0)

{

System.***out***.println(i);

}

}

}

}

Out Put:

1

2

3

5

7

11

13

17

19

23

29

**Even Numbers:**

**public** **class** EvenOrOdd {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**for**(**int** i=1;i<=20;i++)

{

**if**(i%2==0)

System.***out***.println(i);

}

}

}

Out Put:

2

4

6

8

10

12

14

16

18

20

**Fibonacci Series:**

**public** **class** FibonacciSeries {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**int** febCount=10;

**int**[] feb=**new** **int**[febCount];

feb[0]=0;

feb[1]=1;

**for**(**int** i=2;i<febCount;i++)

{

feb[i]=feb[i-1]+feb[i-2];

}

**for**(**int** i=0;i<febCount;i++)

{

System.***out***.println(feb[i]);

}

}

}

Out Put:

0

1

1

2

3

5

8

13

21

34

**Sum of Digits:**

**public** **class** SumOfDigits {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**int** n=1234;

**int** sum=0;

**while**(n!=0)

{

sum=sum+(n%10);

n=n/10;

}

System.***out***.println("Sum of Digits:"+sum);

}

}

Out Put:

Sum of Digits:10

**Common Elements b/w two Arrays:**

**public** **class** CommonElements {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**int** arr1[] = {4,7,3,2,1,5,6,40};

**int** arr2[] = {3,12,9,1,32,40,2};

**for**(**int** i=0;i<arr1.length;i++)

{

**for**(**int** j=0;j<arr2.length;j++)

{

**if**(arr1[i]==arr2[j])

{

System.***out***.println(arr1[i]);

}

}

}

}

}

Out Put:

3

2

1

40

**MaxOne & MaxTwo Elements:**

**public** **class** MaxOneMaxTwo {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**int**[] num = {1,2,3,4,5,6,7,8,9};

**int** maxOne=0;

**int** maxTwo=0;

**for**(**int** n:num)

{

**if**(maxOne<n)

{

maxTwo=maxOne;

maxOne=n;

}

**else** **if**(maxTwo<n)

{

maxTwo=n;

}

}

System.***out***.println("First Max Number:"+maxOne);

System.***out***.println("Second Max Number:"+maxTwo);

}

}

Out Put:

First Max Number:9

Second Max Number:8

**Reading Data From Excel:**

**package** excelExportAndFileIO;

**import** java.io.File;

**import** java.io.FileInputStream;

**import** java.io.IOException;

**import** org.apache.poi.hssf.usermodel.HSSFWorkbook;

**import** org.apache.poi.ss.usermodel.Row;

**import** org.apache.poi.ss.usermodel.Sheet;

**import** org.apache.poi.ss.usermodel.Workbook;

**import** org.apache.poi.xssf.usermodel.XSSFWorkbook;

**public** **class** ReadGuru99ExcelFile {

**public** **void** readExcel(String filePath,String fileName,String sheetName) **throws** IOException{

//Create an object of File class to open xlsx file

File file = **new** File(filePath+"\\"+fileName);

//Create an object of FileInputStream class to read excel file

FileInputStream inputStream = **new** FileInputStream(file);

Workbook guru99Workbook = **null**;

//Find the file extension by splitting file name in substring and getting only extension name

String fileExtensionName = fileName.substring(fileName.indexOf("."));

//Check condition if the file is xlsx file

**if**(fileExtensionName.equals(".xlsx")){

//If it is xlsx file then create object of XSSFWorkbook class

guru99Workbook = **new** XSSFWorkbook(inputStream);

}

//Check condition if the file is xls file

**else** **if**(fileExtensionName.equals(".xls")){

//If it is xls file then create object of XSSFWorkbook class

guru99Workbook = **new** HSSFWorkbook(inputStream);

}

//Read sheet inside the workbook by its name

Sheet guru99Sheet = guru99Workbook.getSheet(sheetName);

//Find number of rows in excel file

**int** rowCount = guru99Sheet.getLastRowNum()-guru99Sheet.getFirstRowNum();

//Create a loop over all the rows of excel file to read it

**for** (**int** i = 0; i < rowCount+1; i++) {

Row row = guru99Sheet.getRow(i);

//Create a loop to print cell values in a row

**for** (**int** j = 0; j < row.getLastCellNum(); j++) {

//Print Excel data in console

System.***out***.print(row.getCell(j).getStringCellValue()+"|| ");

}

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

}

}

//Main function is calling readExcel function to read data from excel file

**public** **static** **void** main(String...strings) **throws** IOException{

//Create an object of ReadGuru99ExcelFile class

ReadGuru99ExcelFile objExcelFile = **new** ReadGuru99ExcelFile();

//Prepare the path of excel file

String filePath = System.*getProperty*("user.dir")+"\\src\\excelExportAndFileIO";

//Call read file method of the class to read data

objExcelFile.readExcel(filePath,"ExportExcel.xlsx","ExcelGuru99Demo");

}

}

**Write Data into Excel:**

**package** excelExportAndFileIO;

**import** java.io.File;

**import** java.io.FileInputStream;

**import** java.io.FileOutputStream;

**import** java.io.IOException;

**import** org.apache.poi.hssf.usermodel.HSSFWorkbook;

**import** org.apache.poi.ss.usermodel.Cell;

**import** org.apache.poi.ss.usermodel.Row;

**import** org.apache.poi.ss.usermodel.Sheet;

**import** org.apache.poi.ss.usermodel.Workbook;

**import** org.apache.poi.xssf.usermodel.XSSFWorkbook;

**public** **class** WriteGuru99ExcelFile {

**public** **void** writeExcel(String filePath,String fileName,String sheetName,String[] dataToWrite) **throws** IOException{

//Create an object of File class to open xlsx file

File file = **new** File(filePath+"\\"+fileName);

//Create an object of FileInputStream class to read excel file

FileInputStream inputStream = **new** FileInputStream(file);

Workbook guru99Workbook = **null**;

//Find the file extension by splitting file name in substring and getting only extension name

String fileExtensionName = fileName.substring(fileName.indexOf("."));

//Check condition if the file is xlsx file

**if**(fileExtensionName.equals(".xlsx")){

//If it is xlsx file then create object of XSSFWorkbook class

guru99Workbook = **new** XSSFWorkbook(inputStream);

}

//Check condition if the file is xls file

**else** **if**(fileExtensionName.equals(".xls")){

//If it is xls file then create object of XSSFWorkbook class

guru99Workbook = **new** HSSFWorkbook(inputStream);

}

//Read excel sheet by sheet name

Sheet sheet = guru99Workbook.getSheet(sheetName);

//Get the current count of rows in excel file

**int** rowCount = sheet.getLastRowNum()-sheet.getFirstRowNum();

//Get the first row from the sheet

Row row = sheet.getRow(0);

//Create a new row and append it at last of sheet

Row newRow = sheet.createRow(rowCount+1);

//Create a loop over the cell of newly created Row

**for**(**int** j = 0; j < row.getLastCellNum(); j++){

//Fill data in row

Cell cell = newRow.createCell(j);

cell.setCellValue(dataToWrite[j]);

}

//Close input stream

inputStream.close();

//Create an object of FileOutputStream class to create write data in excel file

FileOutputStream outputStream = **new** FileOutputStream(file);

//write data in the excel file

guru99Workbook.write(outputStream);

//close output stream

outputStream.close();

}

**public** **static** **void** main(String...strings) **throws** IOException{

//Create an array with the data in the same order in which you expect to be filled in excel file

String[] valueToWrite = {"Mr. E","Noida"};

//Create an object of current class

WriteGuru99ExcelFile objExcelFile = **new** WriteGuru99ExcelFile();

//Write the file using file name, sheet name and the data to be filled

objExcelFile.writeExcel(System.*getProperty*("user.dir")+"\\src\\excelExportAndFileIO","ExportExcel.xlsx","ExcelGuru99Demo",valueToWrite);

}

}