1. Write a program to calculate the factorial of a number using a while loop.

**package** Factorial;

**class** Test\_Factorial{

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

**int** i = 1;

**int** fact=1;

**int** number=5;

**while**(i<=number){

fact=fact\*i;

i++;

}

System.***out***.println("Factorial of "+number+" is: "+fact );

}

}

2. Write a program to print Fibonacci series (10 values).

**package** Fibonacci;

**class** Test\_Fibonacci{

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

**int** first = 0;

**int** second = 1;

**int** third;

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

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

**do**

{

third = first + second;

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

first = second;

second = third;

} **while** (first <= 10);

}

}

3. Write a program to sort the elements of an array in ascending order.

**package** Arrays;

**import** java.util.Arrays;

**class** TestArrays\_ascend{

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

{

**int**[] array\_test = **new** **int**[] { -1, 12, 8, -2, 1, 0 };

System.***out***.print("Elements of original array: ");

**for** (**int** i = 0; i < array\_test.length; i++) {

System.***out***.print(array\_test[i] + " ");

}

Arrays.*sort*(array\_test);

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

System.***out***.println(

"Elements of array sorted in ascending order : "

+ Arrays.*toString*(array\_test));

}

}

4. . Write a program to check whether the current year is leap year or not. Users will enter a year value

**package** Triangle\_array;

**public** **class** Test\_TriangleArray {

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

{

**int** i, j;

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

{

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

{

System.***out***.print("0");

}

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

}

}

}

**package** Leap Year;

**import** java.util.Scanner;

**public** **class** Main {

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

Scanner in = **new** Scanner(System.***in***);

**int** y;

System.***out***.println ("Enter the year");

y = in.nextInt();

**if** (y % 400 == 0) {

System.***out***.println ("\n" + y+ " is the leap year.");

} **else** **if** (y % 100 == 0) {

System.***out***.println ("\n" + y + " is not the leap year.");

} **else** **if** (y % 4 == 0) {

System.***out***.println ("\n" + y + " is the leap year.");

} **else** {

System.***out***.println ("\n" + y + " is not the leap year.");

}

}

}

5. Write a program to print the first 10 prime numbers.

**package** PrimeNumbers;

**public** **class** Prime\_numbers {

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

**int** i =0;

**int** num =0;

String primeNumbers = "";

**for** (i = 1; i <= 10; i++)

{

**int** counter=0;

**for**(num =i; num>=1; num--)

{

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

{

counter = counter + 1;

}

}

**if** (counter ==2)

{

primeNumbers = primeNumbers + i + " ";

}

}

System.***out***.println("Prime numbers from 1 to 10 are :");

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

}

}

6. Write a program to calculate the area of a triangle. Users will enter the values for base and height of the triangle.

**package** Area\_triangle;

**import** java.util.Scanner;

**public** **class** Area\_triangle {

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

@SuppressWarnings("resource")

Scanner scanner = **new** Scanner(System.***in***);

System.***out***.println("Enter the base of the Triangle:");

**double** base = scanner.nextDouble();

System.***out***.println("Enter the height of the Triangle:");

**double** height = scanner.nextDouble();

**double** area = (base \* height)/2;

System.***out***.println("Area of Triangle is: " + area);

}

}

7. Write a program to print the sum of the first 20 natural numbers

**package** Sum\_Naturalnumber;

**public** **class** Sum\_Naturalnumber {

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

{

**int** i, num = 10, sum = 0;

**for**(i = 1; i <= num; ++i)

{

sum = sum + i;

}

System.***out***.println("Sum of First 10 Natural Numbers is = " + sum);

}

}

8. Write a program to reverse the elements of an array where the array size as well as the array values are entered by the user.

**package** Array\_reverse;

**import** java.util.Scanner;

**public** **class** Array\_reverse {

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

{

**int** n, i,j=0;

@SuppressWarnings("resource")

Scanner s = **new** Scanner(System.***in***);

System.***out***.print("Enter number of elements in the array:");

n = s.nextInt();

**int** array[] = **new** **int**[n];

**int** rev[] = **new** **int**[n];

System.***out***.println("Enter "+n+" elements ");

**for**( i=0; i < n; i++)

{

array[i] = s.nextInt();

}

System.***out***.println("Reverse of an array is :");

**for**( i=n;i>0 ; i--,j++)

{

rev[j] = array[i-1];

System.***out***.println(rev[j]);

}

}

}

9. Write a program to print only even numbers till 50.

**package** EvenNumber;

**public** **class** Test\_EvenNumber {

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

**int** Numbers = 50;

System. ***out***. println("Printing Even numbers between 1 and " + Numbers);

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

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

System.***out***.print(i + " ");

}

}

}

}

10. 10. Write a program to print this output using a two-dimensional array.

Triangle Array 0 00 000 0000 00000 000000 0000000 00000000 000000000 000000000

**package** Triangle\_array;

**public** **class** Test\_TriangleArray {

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

{

**int** i, j;

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

{

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

{

System.***out***.print("0");

}

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

}

}

}

**Assignment 2**

1. Java Program to Calculate average of numbers using Array Example: values [] = 1, 2,3 4, 5 average = 3

**Package** ArrayAverage;

**Public class** ArrayAverage {

public static void main(String[] args) {

int[] array = { 1, 2, 3, 4, 5 };

int length = array.length;

int sum = 0;

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

sum += array[i];

}

double average = sum / length;

System.out.println("Average of array : "+average);

}

}

1. Find all the odd numbers from 79 to 187

**Package** odd numbers;

**public class** odd numbers {

public static void main(String args[]) {

System.out.println("The Odd Numbers are:");

for (int i = 79; i <= 187; i++) {

if (i % 2 != 0) {

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

}

}

}

}

1. Is 13 a prime number?

**Package** Checkprimenumber;

**public class** Checkprimenumber {

static void checkPrime(int n){

int i, m=0, flag=0;

m=n/2;

if(n==0||n==1){

System.out.println(n+" is not prime number");

}

Else

{

for(i=2;i<=m;i++){

if(n%i==0){

System.out.println(n+" is not prime number");

flag=1;

break;

}

}

if(flag==0) { System.out.println(n+" is prime number"); }

}

}

public static void main(String args[]){

checkPrime(13);

}

}

1. Find the sum of digits Example: int n = 1234; output: 10

**Package** Sumofnumbers;

**public class** sumofnumbers

{

static int findSum(int number)

{

int sum = 0;

while (number != 0)

{

sum = sum + number % 10;

number = number/10;

}

return sum;

}

public static void main(String args[])

{

int number = 1234;

System.out.println("The sum of digits: "+findSum(number));

}

}

1. Reverse a number 123

**Package** reversenumber ;

i**mport** java.util.Scanner;

**public class** reversenumber

{

public static void reverseNumber(int number)

{

if (number < 10)

{

System.out.println(number);

return;

}

else

{

System.out.print(number % 10);

reverseNumber(number/10);

}

}

public static void main(String args[])

{

System.out.print("Enter the number that you want to reverse: ");

Scanner sc = new Scanner(System.in);

int num = sc.nextInt();

System.out.print("The reverse of the given number is: ");

reverseNumber(num);

}

}

1. Find the duplicates in an array 12, 32, 12, 45, 65, 93, 0, 23, 45, 6

**Package** Duplicatesinarray;

I**mport** java.util.HashSet;

i**mport** java.util.Set;

p**ublic** class Duplicatesinarray {

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

int arr[] = {12, 32, 12, 45, 65, 93, 0, 23, 45, 6};

Set arrSet = new HashSet<>();

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

if (arrSet.contains(arr[i])) {

System.out.println(arr[i]);

}

else {

arrSet.add(arr[i]);

}

}

}

}

Assignment 3

1. Write a program to print the occurrence of each character in the String “DevLabs Alliance Training”

**Package** occurenceOfCharacter;

i**mport** java.util.HashMap;

**public** class occurenceOfCharacter {

public static void main(String[] args) {

String str = "DevLabs Alliance Training";

HashMap <Character, Integer> hMap = new HashMap<>();

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

if (hMap.containsKey(str.charAt(i))) {

int count = hMap.get(str.charAt(i));

hMap.put(str.charAt(i), ++count);

} else {

hMap.put(str.charAt(i),1);

}

}

System.out.println(hMap);

}

}

1. Write a program to check if a given string is a palindrome or not. Palindrome example: trurt

**Package** Palindrome;

p**ublic** class Palindrome{

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

{

String str, rev = "";

str="trurt";

int length = str.length();

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

rev = rev + str.charAt(i);

if (str.equals(rev))

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

else

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

}

}

1. Write a program to check “brown” is present in the string: A brown fox ran away fast

i**mport** java.util.\*;

**Package** wordpresent;

public class wordpresent {

public static void main(String []args){

String str="A brown fox ran away fast";

String str1="brown";

String x=find\_str(str,str1);

System.out.println(x);

}

public static String find\_str(String sentence,String search) {

int i;

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

for(i=0;i<s.length;i++)

{

if(search.equals(s[i]));

}

return search +" present in "+sentence ;

}

}

1. Write a program to convert String to a character array and character array to String.

**Package** chararray;

public class chararray {

public static void main(String[] args)

{

char[] charArray = { 'J', 'a', 'v', 'a' };

String str = new String(charArray);

System.out.println(str);

char[] ch = new char[str.length()];

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

ch[i] = str.charAt(i);

}

for (char c : ch)

{

System.out.println(c);

}

1. Write a program to throw NumberFormatException and handle it appropriately with a proper message. If you pass invalid input to parseInt(str), this exception will be thrown.

**Package** NumberFormatException;

public class NumberFormatException {

private static final String inputString = "123.33";

public static void main(String[] args) {

try {

int a = Integer.parseInt(inputString);

}catch(NumberFormatException ex){

System.err.println("Invalid string in argumment");

}

}

}

1. Write a program where a method declares that it throws ArithmeticException

**Package** TestArthemetic;

import java.util.Scanner;

class TestArthemetic {

public static void main(String[] args) {

Scanner s = new Scanner(System.in);

System.out.println("Please enter your roll number");

int roll = s.nextInt();

try {

if (roll < 0) {

throw new ArithmeticException(

"The number entered is not positive"

);

} else {

System.out.println("Valid roll number");

}

} catch (ArithmeticException e) {

System.out.println("An exception is thrown");

System.out.println(e.getMessage());

}

}

1. Write a program with nested try blocks.

**Package** Nestedtry;

public class Nestedtry {

public static void main(String args[]) {

try {

int a[]=new int[10];

// displaying element at index 12

System.out.println(a[12]);

// another try block

try {

System.out.println("Division");

int res = 100/ 0;

}

catch (ArithmeticException ex2) {

System.out.println("Sorry! Division by zero isn't feasible!");

}

}

catch (ArrayIndexOutOfBoundsException ex1) {

System.out.println("ArrayIndexOutOfBoundsException");

}

}

}

1. Write a program to re-throw an exception. (throw inside catch block)

**Package** Rethroweception;

**public** class Rethroweception

{

static void divide()

{

int x,y,z;

try

{

x = 6 ;

y = 0 ;

z = x/y ;

System.out.println(x + "/"+ y +" = " + z);

}

catch(ArithmeticException e)

{

System.out.println("Exception Caught in Divide()");

System.out.println("Cannot Divide by Zero in Integer Division");

throw e; // Rethrows an exception

}

}

public static void main(String[] args)

{

System.out.println("Start of main()");

try

{

divide();

}

catch(ArithmeticException e)

{

System.out.println("Rethrown Exception Caught in Main()");

System.out.println(e);

}

}

}

Assignment 4

1. Find duplicate characters with their occurrences count using HashMap.

**package** DuplicateCharacters;

**import** java.util.HashMap;

**import** java.util.Map;

**public** class L4P1DuplicateCharacters {

static String text;

**public** static void countDuplicateCharacters(String str) {

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

char[] charArray = str.toCharArray();

for (char c : charArray) {

if (map.containsKey(c)) {

int getMapeginIndex = map.get(c);

map.put(c, getMapeginIndex + 1);

} else {

map.put(c, 1);

}

}

for (Map.Entry<Character, Integer> entry : map.entrySet()) {

if (entry.getValue() > 1) {

System.out.println(entry.getKey() + " : " + entry.getValue());

}

}

}

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

text = "geeksforgeeks";

countDuplicateCharacters(text);

}

}

2.Reverse an Arraylist.

p**ackage** ReverseArrayList;

i**mport** java.util.ArrayList;

i**mport** java.util.Collections;

**import** java.util.List;

**public** class ReverseArrayList {

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

{

List<String> mylist = new ArrayList<String>();

mylist.add("practice");

mylist.add("code");

mylist.add("quiz");

mylist.add("geeksforgeeks");

System.out.println("Original List : " + mylist);

Collections.reverse(mylist);

System.out.println("Modified List: " + mylist);

}

}

}

1. Check if a particular key exists in HashMap.

**package** KeyHashmap;

**import java.util.HashMap;**

**public** class KeyHashmap {

p**ublic** static void main(String[] args) {

int keyToBeChecked = 11;

HashMap<Integer, String> hashmap = new HashMap<Integer, String>();

// Adding Key and Value pairs to HashMap

hashmap.put(11,"Sudha");

hashmap.put(22,"Karthik");

hashmap.put(33,"Naveen");

hashmap.put(44,"Dilip");

hashmap.put(55,"ravi");

boolean flag = hashmap.containsKey(keyToBeChecked);

System.out.println("Does Key "+keyToBeChecked+" exists in HashMap : " + flag);

}

}

1. Convert keys of a map to a list.

p**acka**ge MapKeytoList;

i**mport** java.util.ArrayList;

i**mport** java.util.HashMap;

**import** java.util.Map;

**public** class MapKeytoList {

public static void main(String args[]){

Map<Integer, String> myMap = new HashMap<>();

myMap.put(1, "JavaX");

myMap.put(2, "JavaFX");

myMap.put(3, "CoffeeScript");

myMap.put(4, "TypeScript");

ArrayList<Integer> keyList = new ArrayList<Integer>(myMap.keySet());

ArrayList<String> valueList = new ArrayList<String>(myMap.values());

System.out.println("contents of the list holding keys the map ::"+keyList);

System.out.println("contents of the list holding values of the map ::"+valueList);

}

}

1. Copy all elements of a HashSet to an Object array

**package** ElemHastSet2ObjectArray;

**impor**t java.util.HashSet;

**import** java.util.Set;

**public** class ElemHastSet2ObjectArray {

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

Set<Integer> hs = new HashSet<Integer>();

hs.add(15);

hs.add(71);

hs.add(82);

hs.add(89);

hs.add(91);

hs.add(93);

hs.add(97);

hs.add(99);

System.out.println("Elements in set = " + hs);

System.out.println("Copying all elements...");

Object[] obArr = hs.toArray();

for (Object ob : obArr)

System.out.println(ob);

}

}

1. Get highest and lowest value stored in TreeSet

package com. HighestLowestTreeSet;

import java.util.TreeSet;

public class GetHighestLowestTreeSet {

public static void main(String args[]) {

TreeSet<String> tSet = new TreeSet<String>();

tSet.add("1");

tSet.add("3");

tSet.add("2");

tSet.add("5");

tSet.add("4");

System.out.println(" Highest value Stored in Java TreeSet is : " + tSet.last());

System.out.println(" Lowest value Stored in Java TreeSet is : " + tSet.first());

}

}

1. Sort ArrayList of Strings alphabetically

package SortArrayListOfStrings;

import java.util.Arrays;

import java.util.Comparator;

import java.util.List;

public class SortArrayListOfStrings {

public static void main(String[] args) {

List<String> names = Arrays.asList("Alex", "Charles", "Brian", "David");

//1. Natural order

names.sort( Comparator.comparing( String::toString ) );

System.out.println(names);

//2. Reverse order

names.sort( Comparator.comparing( String::toString ).reversed() );

System.out.println(names);

}

}

1. Get Set view of keys from HashTable.

package com.labexcersice;

import java.util.Enumeration;

import java.util.Hashtable;

import java.util.Iterator;

import java.util.Set;

public class L4P8GetSetOfHashSets {

public static void main(String[] args) {

Hashtable<String, String> hashTable = new Hashtable<String, String>();

hashTable.put("1", "One");

hashTable.put("2", "Two");

hashTable.put("3", "Three");

Set st = hashTable.keySet();

Iterator itr = st.iterator();

while (itr.hasNext()) {

System.out.println(itr.next());

}

Enumeration e = hashTable.keys();

while (e.hasMoreElements()) {

System.out.println(e.nextElement());

}

}

}