**DAY 2 – JAVA ASSIGNMENT PROBLEMS SOLUTION**

1. **Can you print below diagram?**

\*

\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*\*

1. **Can you print first 10 prime numbers?**

|  |
| --- |
| public class PrimeNumberCount {  public static void main(String[] args) {  System.out.println("Prime Number List");  System.out.println(" "+ "1");  int flag = 0;  int count = 3;  int primenumbercount =2;  System.out.println("Entering for loop.");    do {  int numberdividebytwo = (count / 2);  for (int i=2; i< count; i++)  { if (count %i == 0 | (count %2 == 0 && count != 2) | (count %3 == 0 && count !=3) | (count %5 == 0 && count !=5) | (count %7 == 0 && count!=7)) { break; }  flag =1;}    if (flag ==1) {  System.out.println(count);  primenumbercount++;  flag=0;}    if (primenumbercount==10) {  break;}    count++;  }while (count<=1000); }    }  Output: Prime Number List  1 2  Entering for loop.  3 5 7 11 13 17 19 23  try {File file = new File("./FirstTenPrimeNumbers.txt");  // if file doesnt exists, then create it if (!file.exists()) { file.createNewFile();  c) Write to a file  FileWriter fw = new FileWriter(file.getAbsoluteFile());  BufferedWriter bw = new BufferedWriter(fw);  bw.write("File input line one.");  for (int i=0; i< device\_ids.size(); i++)  { bw.write("File input line two."); }  bw.close();System.out.println("Done");}  catch (IOException e) { e.printStackTrace();  } |

1. **Three Ways of specifying file directories**

|  |
| --- |
| public interface Variables {  static final String PM\_BUILD\_WORKSPACE = System.getenv("PM\_WORKSPACE");  1) String csvFile = TEST\_DATA\_CSV\_FILE; // From system properties  2) // String csvFile = "../data/TestData.csv"; // One folder down  3) // String csvFile = "./data/TestData.csv"; // Current folder |

1. **Please implement a function that accepts two parameters say Parameter1 & Parameter2 and gives the output of the number of times the parameter1 occurs in parameter2**

|  |
| --- |
| // Reading an Integer  public static void main (args[]) int a = args[0];  // Reading a single string  Scanner SC = new Scanner(System.in);  System.out.println("Enter parameter1:");  String parameter1 = SC.next();// reading a statement  BufferedReader reader = new BufferedReader(new InputStreamReader(System.in)); System.out.println("Enter parameter2:");  String parameter2= reader.readLine();  // Converting parameter2 into scanner object  Scanner SCParameter2 = new Scanner(parameter2);  // Storing words of parameter2 in a linked list.  LinkedList words = new LinkedList();  while (SCParameter2.hasNext()) { words.add(SCParameter2.next()); }  // Iterating through the linked list and counting number of occurrences of parameter1 in parameter2.  Iterator LLIterator = words.iterator();  int count =0; while (LLIterator.hasNext()) { if(parameter1.equalsIgnoreCase(LLIterator.next())) { count++; } } System.out.println("The Number of occurances of parameter1 in parameter2 is::" + count); |

1. **Given an array of integers, sort the integer values.**

|  |
| --- |
| package Assignment;  // import java.io.System;  public class BubbleSort{public static void main(String args[]) {  int i; int array[] = {12,9,4,99,120,1,3,10};// prints the value before sorting array.  System.out.println("Values Before bubble sort of Integers:\n");  for(i = 0; i < array.length; i++)  System.out.print( array[i]+" "); System.out.println();  // sorting array  bubble\_srt(array, array.length);  // printing the elements of array after the sort  System.out.print("Values after the sort:\n");  for(i = 0; i < array.length ; i++)  System.out.print(array[i]+" ");  System.out.println();}  // end of main  // static bubble sort method  public static void bubble\_srt( int a[], int n )  {int i, j,t=0;  for (i = 0; i < n; i++) {  // since highest value is put at the last in first iteration  for (j = 1; j < n-i; j++) {  if(a[j-1] > a[j]) { t = a[j-1]; a[j-1]=a[j]; a[j]=t; } } } }  // end of bubble\_srt() }// end of class  Output Values Before bubble sort of Integers:12 9 4 99 120 1 3 10 Values after the sort: 1 3 4 9 10 12 99 120 |

1. **Given an array of integers, print only odd numbers.**

|  |
| --- |
| package Assignment;  public class OddNumbers {  public static void main (String args[]) {  int i; int array[] = {12,9,4,99,120,1,3,10};  // print the elements of array  System.out.print("Elements of the array are ::\n");  for(i = 0; i < array.length; i++)  System.out.print( array[i]+" ");  System.out.println(); System.out.println();  // logic for printing the odd elements of the array System.out.println("Printing the odd numbers of the array::");  for (i=0;i < array.length;i++ ) { if (array[i] % 2 != 0 ) { System.out.print(array[i]+" "); } else continue;}  Output  Elements of the array are 12 9 4 99 120 1 3 10Printing the odd numbers of the array 9 99 1 |

1. **Given an array of integers move all even numbers to the beginning of the array.**

|  |
| --- |
| package Assignment;  public class MoveEven { public static void main (String args[]) {  int i; int array[] = {12,9,4,99,120,1,3,10};// The array elements before moving even elements  System.out.println("Values Before moving even integers front of array\n"); for(i = 0; i < array.length; i++)  System.out.print( array[i]+" ");  System.out.println();  // Function which moves the even elements to the front of the array. move(array, array.length);  // Printing the array elements after the even integers are moved to front. System.out.println("Values After moving even integers front of array\n"); for(i = 0; i < array.length; i++)  System.out.print( array[i]+" ");  System.out.println(); }  public static void move (int a[],int n) {  int i,j,t; for(i = 0; i < n; i++) {  if (a[i]%2 ==0) { for (j = i; j > 0; j--) {  t = a[j-1]; a[j-1]=a[j]; a[j]=t; } } } }}  Output Values Before moving even integers front of array12 9 4 99 120 1 3 10 Values After moving even integers front of array10 120 4 12 9 99 1 |

1. **Print the unique numbers and also print the number of occurrences of duplicate numbers in a two dimensional array.**

|  |
| --- |
| package Assignment;  // This can be accomplished using hash tables. Please look at it.  public class Unique {public static void main(String[] args)  {int i; int array[] = {12,9,4,99,120,1,3,10, 12, 4,4, 120,3,3,3};  // Printing the array elements int limit = array.length; System.out.println("Printing the elements of array\n");  for(i = 0; i < array.length; i++)  System.out.print( array[i]+" ");  System.out.println();// initializing a two dimensional array. int record[][]= new int [limit][2];// filling the two dimensional array.  for (i=0;i<array.length;i++) {  record[i][0] = 0; record[i][1] = 0; }  int flag;// For pasting the unique elements into another array.  for (i=0;i<array.length;i++) { flag =1; for (int j=0;j<i;j++ )  { if (array[i] == array[j]) { flag ++; } }  if (flag == 1) {  holder[i][0] = array[i]; } }  // For counting the number of occurrences.  int j; flag=1; for (i=0;i<holder.length;i++) {  flag=0; for (j=i;j<array.length;j++) {  if (record[i][0] == array[j]) { flag++; } }  // Assigning flag value to the holder.  record[i][1]= flag;}  // Printing the unique elements and number of their occurrences in 2D array. System.out.println("Printing the unique elements as 2D array");  for (i=0;i<record.length;i++) {  System.out.println(record[i][0]+" "+record[i][1]);}} }  Output Printing the elements of array12 9 4 99 120 1 3 10 12 4 4 120 3 3 3 Printing the unique elements as 2D array 12 2 9 1 4 3 99 1 120 2 1 1 3 4 10 |

1. **Given an array of integers check the Fibonacci series.**

|  |
| --- |
| package Assignment;  import java.util.Scanner;  public class Fibonnacci {static int n; static int [] fibo;  static Scanner console=new Scanner (System.in);  public static void main(String[] args) {  // Array for testing whether it is fibonacci or not.  int check[] = {1,1,2,3,5,8};  // Printing the given array elements.  System.out.println("Printing the given array::");  for (int j=0;j<check.length;j++)  { System.out.print(check[j]+" "); }  System.out.println();n = check.length;  // Generating new array containing fibonnaci numbers.  fibo = new int [n];  fill(fibo,n);  boolean flag = true;  int i; for (i=0;i<n;i++) { if (fibo[i] != check[i]) {  flag = false; break; } }  if (flag) {  System.out.println("The given array elements form fibonnacci series."); } else {  System.out.println("The given array elements do not form fibonnacci series."); } }  // Logic for generating Fibonnacci numbers.  public static void fill(int[]fibo, int n) {  int i =0; fibo[0] = 1; fibo[1] = 1;  for (i=2;i<n;i++) fibo[i]=fibo[i-1]+fibo[i-2];  }// end of fill function  }// end of class  Output Printing the given array 1 1 2 3 5 8 The given array elements form fibonnacci series. |

1. **Given an array of integers check the Palindrome of the series**

|  |
| --- |
| package Assignment;  public class Palindrome {  public static void main(String[] args) {  int i; int array[] = {1,3,4,3,1};  // prints the value before sorting array.  System.out.println("Elements of the array:");  for(i = 0; i < array.length; i++)  System.out.print( array[i]+" ");  // System.out.println();  int flag =1;  for (i=0; i< (array.length/2);i++)  { if (array[i] != array[array.length -i-1] )  { flag = 0; } }  System.out.println();  System.out.println();  if (flag == 0) {  System.out.println("It is not a palindrome");}  else { System.out.println("It is a palindrome."); }}}  Output Elements of the array: 1 3 4 3 1Its a palindrome. |

1. **Given a string print the unique words of the string.**

|  |
| --- |
| package Assignment;  import java.util.HashSet;  import java.util.Iterator;  import java.util.Scanner;  import java.util.Set;  // import java.util.TreeSet;  // Implementing sets to find the unique words.  public class UniqueWord {  public static void main(String[] args) {  // Hash Set implementing the Set.  Set words = new HashSet();  // using a sample string to print the unique words  String sample ="This is a test is a test a test test";  Scanner in = new Scanner(sample);  //System.out.println("Please enter the string");  while (in.hasNext()) {String word = in.next();  // using the add function to add the words into the hash set.  words.add(word);}  // Used for moving through the set and printing the words.  Iterator iter = words.iterator();  // Printing the unique words of the string.  System.out.println("Printing the unique words of the given string::");  for (int i = 1; i <= 20 && iter.hasNext(); i++)  // using iterator function to read the elements of hash set. System.out.print(" "+ iter.next()+ " "); } }  Output: Printing the unique words of the given string:: is test a This |

1. **Given a string print the reverse of the string.**

|  |
| --- |
| package Assignment;  // program for printing the reverse of the string.  public class StringReverse { public static void main(String[] args) {  String str = "molecule"; String reverse ="";  int i=0; // printing the original string  System.out.println("Original String:: "+ str); // converting string to character array. char rev[] = str.toCharArray();  // appending characters to reverse string.  for (i=rev.length-1;i>=0;i--) {  reverse = reverse + rev[i]; }  System.out.println();  // Printing the reversed String  System.out.println("Reversed String is:: " + reverse); } }  Output Original String:: moleculeReversed String is::  Elucelom |

1. **Read a file content and write it to a new file in reverse order.(reverse line 1-10 to line 10-1)**

|  |
| --- |
| package Assignment;  import java.io.\*;  import java.util.LinkedList;  public class ReverseFile {  public static void main(String args[]) {  try{// using fileinputstream to read contents inputFile.txt  FileInputStream fstream = new FileInputStream("C:\\Users\\Sunil Kumar\\Desktop\\White Box Training\\Java Programs\\Assignment\\inputFile.txt");  DataInputStream in = new DataInputStream(fstream);  BufferedReader br = new BufferedReader(new InputStreamReader(fstream));  String strLine;  // using LinkedList to store the lines in the file.  LinkedList list = new LinkedList();  //Reading input file line by line  while ((strLine = br.readLine()) != null) {  list.add(strLine); }  // Opening the outPut.txt file using FileWriter.  FileWriter filestream = new FileWriter("C:\\Users\\Sunil Kumar\\Desktop\\White Box Training\\Java Programs\\Assignment\\outputFile.txt");  BufferedWriter out = new BufferedWriter(filestream);  // Writing the lines in reverse fashion into outputFile.txt  int i; int len = list.size();  for (i=len-1;i>=0;i--) {  out.write(list.get(i));  out.write("\n"); }  out.close();  in.close(); }  catch (Exception e){  //Catch exception if any  System.err.println("Error: " + e.getMessage()); } }}  Input  File.txt  First line.  Second Line.  Third Line.  Fourth Line.  Fifth Line.  Sixth Line.  Seventh Line.  Eighth Line.  Ninth Line.  Tenth Line.  outputFile.txt  Tenth Line.  Ninth Line.  Eighth Line.  Seventh Line.  Sixth Line.  Fifth Line.  Fourth Line.  Third Line.  Second Line.  First line. |

1. **Write a java program which provides API for database "select" and "Update" package MySQL;**

|  |
| --- |
| import java.sql.\*;  public class DBAccess {  public static Statement stmt = null;  public static Connection conn = null;  public ResultSet execute(String query) {  ResultSet rs = null;try {  // Execute the query and return the ResultSet  stmt.executeQuery(query);  rs = stmt.getResultSet(); }  catch (Exception e) {  System.err.println ("Cannot connect to database server"); System.out.println(e);  //e.printStackTrace();  }  return rs; }  public int update(String query) {  int count = 0;  try {  count = s.executeUpdate(query); }  catch (Exception e) {  System.err.println("Cannot conect to database server."); System.out.println(e); }  return count; }  public static void main (String[] args) {  //Connection conn = null;DBAccess db = new DBAccess();  try { String userName = "root";  String password = "good";  // localhost - Name of the server.  String url = "jdbc:mysql://localhost/test";  // Create one driver instance and create one or more connection instances.  // Standard syntax of creating instance of singleton class.  Class.forName ("com.mysql.jdbc.Driver").newInstance ();  // Connection instance using the Driver.  conn = DriverManager.getConnection (url, userName, password);  System.out.println ("Database connection established");  stmt = conn.createStatement ();  int count;  // Two types of methods present in the JDBC code - executeUpdate and // executeQuery  // Passing the query and updating the record.  String query2 = "Update EMP set email='hare@gmail.com' where id = 2;";  count = db.update(query2);  System.out.println("Updated record count = " + count);  // Passing query and s executing query and returning rs. String query1 = "select \* from EMP";  ResultSet rs = db.execute(query1);  while (rs.next ()) {  int idVal = rs.getInt ("id"); String nameVal = rs.getString ("name");  String catVal = rs.getString ("email");  System.out.println ( "id = " + idVal + ", name = " + nameVal + ", email = " + catVal); ++count; }  rs.close (); s.close ();s.close ();  // System.out.println (count + " rows were inserted");}  catch (Exception e) {  System.err.println ("Cannot connect to database server"); System.out.println(e); e.printStackTrace(); }  finally { if (conn != null) { try { conn.close ();  System.out.println ("Database connection terminated"); }  catch (Exception e) { /\* ignore close errors \*/ } } } } }  Output  Database connection established  Updated record count = 1  id = 1, name = Abcd, email = [abcded@gmail.com](mailto:abcded@gmail.com)  id = 2, name = Sunil, email = [abc@gmail.com](mailto:abc@gmail.com)  id = 1, name = Sunil2, email = [abcdef@gmail.com](mailto:abcdef@gmail.com)  Database connection terminated |