

## SBA-1 Milestone-Helani Paul 210914

1. Write a Java Program to reverse a string without using String inbuilt function reverse ().

```
import java.util.Scanner;

public class ReverseString {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the string: ");
        String str = sc.nextLine();
        String temp = " ";

        for(int i=0;i<str.length();i++)
        {
            char ch = str.charAt(i);
            temp = ch+temp;
        }
        System.out.println("Reversed String is : "+temp);
        sc.close();
    }
}
```

Output:

```
Enter the string:
good morning
Reversed String is : gninrom doog
```

program to take an input number from the programmer and calculate all the prime numbers from 0 to that number. Store all the prime numbers in an array and display the array elements. Example: Input=10 Output:

2. Write a

1,2,3,5,7

```
import java.util.Scanner;

public class PrimeUpton {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the limit: ");
        int n = sc.nextInt();
        int count;
        int len=0;
        int[] arr = new int[n];

        for(int i=2;i<=n;i++)
        {
            count = 0;
            for(int j=2;j<=i/2;j++)
            {
                if(i%j == 0)
                {
                    count++;
                    break;
                }
            }
            if(count == 0)
            {
                arr[len]=i;
                len++;
            }
        }
        System.out.println("The Array elements are: ");
        for(int x=0;x<len;x++)
        {
            System.out.print(arr[x]+" ");
        }
        sc.close();
    }

}
```

Output:

```
Enter the limit:
10
The Array elements are:
2 3 5 7
```

whether a string or number is palindrome or

not.

Note: input can be a number or a String.

### 3. Write a Java Program to find

```
import java.util.Scanner;
public class PalindromeNumberOrString {
    static boolean isPalindrome(String str)
    {
        int x=0,y=str.length()-1;
        while(x<y)
        {
            if(str.charAt(x)!=str.charAt(y))
            {
                return false;
            }
            x++;
            y--;
        }
        return true;
    }
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the String");
        String str1 = sc.next();
        System.out.println("Enter a number");
        String num = sc.next();
        if(isPalindrome(str1))
        {
            System.out.println(str1+" Is a palindrome");
        }
        else
        {
            System.out.println(str1+" Not a palindrome");
        }
        if(isPalindrome(num))
        {
            System.out.println(num+" Is a palindrome");
        }
        else
        {
            System.out.println(num+" Not a palindrome");
        }

        sc.close();
    }
}
```

Output:

```
Enter the String
malayalam
Enter a number
156
malayalam Is a palindrome
156 Not a palindrome
|
```

4. Write a Java Program to find

the duplicate characters in a string.

```
import java.util.Scanner;

public class DuplicateElements {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the String: ");
        String st = sc.nextLine();
        char[] ar = st.toCharArray();
        System.out.println("The Duplicate elements are: ");
        for(int i=0;i<ar.length;i++)
        {
            int count=0;
            for(int j=i+1;j<ar.length;j++)
            {
                if(ar[i] == ar[j] && ar[i]!=' ')
                {
                    count++;
                    ar[j]='0';
                }
            }
            if(count>0 && ar[i]!='0')
            {
                System.out.print(ar[i]+",");
            }
        }
        sc.close();
    }
}
```

Output:

```
Enter the String:
i am alone in playground
The Duplicate elements are:
i,a,l,o,n,
```

5. Write a Java Program to find

the second-highest number in an array.

```
import java.util.Scanner;

public class SecondLargest {

    public static int getSecond(int[] a,int n)
    {
        int temp;
        for(int i=0;i<n;i++)
        {
            for(int j=i+1;j<n;j++)
            {
                if(a[i]>a[j])
                {
                    temp=a[i];
                    a[i]=a[j];
                    a[j]=temp;
                }
            }
        }
        return a[n-2];
    }

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the array size: ");
        int len = sc.nextInt();
        int[] arr1 = new int[len];
        System.out.println("Enter the array elements: ");
        for(int i=0;i<len;i++)
        {
            arr1[i] = sc.nextInt();
        }
        System.out.println("Second Largest element in the array is: "+getSecond(arr1,len));
        sc.close();
    }

}
```

Output:

```
Enter the array size:
6
Enter the array elements:
4
9
1
7
2
8
Second Largest element in the array is: 8
```

6. Write a java program to

subtract two matrices. Take the input of the matrices from the user.

```
import java.util.Scanner;
public class SubtractMatrix {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter no.of row for array");
        int row = sc.nextInt();
        System.out.println("Enter no.of column for array");
        int col = sc.nextInt();
        int[][] arr1 = new int[row][col];
        int[][] arr2 = new int[row][col];
        int[][] arr3 = new int[row][col];
        System.out.println("Enter the values for array 1 : ");
        for(int i=0;i<row;i++)
        {
            for(int j=0;j<col;j++)
            {
                arr1[i][j] = sc.nextInt();
            }
        }
        System.out.println("Enter the values for array 2 : ");
        for(int i=0;i<row;i++)
        {
            for(int j=0;j<col;j++)
            {
                arr2[i][j] = sc.nextInt();
            }
        }
        for(int i=0;i<row;i++)
        {
            for(int j=0;j<col;j++)
            {
                arr3[i][j] = arr1[i][j] - arr2[i][j];
            }
        }
        System.out.println("The subtract of two arrays : ");
        for(int i=0;i<row;i++)
        {
            for(int j=0;j<col;j++)
            {
                System.out.print(" "+arr3[i][j]+" ");
            }
            System.out.println("");
        }
        sc.close();
    }
}
```

7. Write a Java Program to find



Output:

```
Enter no.of row for array
2
Enter no.of column for array
2
Enter the values for array 1 :
8
6
4
2
Enter the values for array 2 :
3
2
1
0
The subtract of two arrays :
5 4
3 2
```

a string input from the user. Convert all the  
uppercase letters to lowercase and vice-versa.  
Example:  
Input:  
MiCroSoft Output:  
miCROsOFT



7. Write a java program to take

```
import java.util.Scanner;

public class LowertoUpper {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the string: ");
        String str2 = sc.next();
        char ch;
        //Upper case to Lower case
        for(int i=0;i<str2.length();i++)
        {
            ch = str2.charAt(i);
            if(Character.isUpperCase(ch))
            {
                System.out.print(Character.toLowerCase(ch));
            }
            else
            {
                System.out.print(Character.toUpperCase(ch));
            }
        }
        sc.close();
    }
}
```

Output:

```
Enter the string:
MicroSoft
miCRoSOfT
```

using insertion sort.

an input array of integers and sort the array

8. Write a java program to take

```
import java.util.Arrays;
import java.util.Scanner;
public class InsertionSortArray {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the array size");
        int size = sc.nextInt();
        int[] arr1 = new int[size];

        System.out.println("Enter the elements for array : ");
        for(int i=0;i<size;i++)
        {
            arr1[i] = sc.nextInt();
        }
        System.out.println("Before sorting: ");
        System.out.println(Arrays.toString(arr1));
        System.out.println();
        int current=0,j=0;

        for(int i=0;i<size;i++)
        {
            current = arr1[i];
            j = i-1;
            while(j>=0 && arr1[j]>current)
            {
                arr1[j+1] = arr1[j];
                j--;
            }
            arr1[j+1]=current;
        }
        System.out.println();
        System.out.println("After Insertion sorting: ");
        System.out.println(Arrays.toString(arr1));
        sc.close();
    }
}
```

Output:

```
Enter the array size
8
Enter the elements for array :
24
12
6
2
14
42
10
5
Before sorting:
[24, 12, 6, 2, 14, 42, 10, 5]

After Insertion sorting:
[2, 5, 6, 10, 12, 14, 24, 42]
```

an input array of integers and search for a particular number given by the user. Use binary search algorithm.

9. Write a java program to take

```
import java.util.Scanner;
public class BinarySearchArray {
    static int binarySearchmethod(int a[],int beg,int end,int val)
    {
        int mid;
        if(end>=beg)
        {
            mid=(beg + end)/2;
            if(a[mid] == val)
            {
                return mid+1;
            }
            else if(a[mid] < val)
            {
                return binarySearchmethod(a,mid+1,end,val);
            }
            else
            {
                return binarySearchmethod(a,beg,mid-1,val);
            }
        }
        return -1;
    }
}
```

```
public static void main(String[] args) {
    // TODO Auto-generated method stub
    Scanner sc = new Scanner(System.in);
    System.out.println("Enter the array size");
    int size = sc.nextInt();
    int[] arr = new int[size];
    System.out.println("Enter the elements for sorted array : ");
    for(int i=0;i<size;i++)
    {
        arr[i] = sc.nextInt();
    }
    System.out.println("Enter the element to be searched: ");
    int item = sc.nextInt();
    System.out.println();
    int result = binarySearchmethod(arr,0,size-1,item);
    if(result == -1)
    {
        System.out.println("Element "+item+" is not present in the array");
    }
    else
    {
        System.out.println("Element "+item+" is present at "+result+" position in the array");
    }
    sc.close();
}
```

```
Enter the array size
8
Enter the elements for sorted array :
5
12
17
23
25
39
51
89
Enter the element to be searched:
51
|
Element 51 is present at 7 position in the array
```

10. Write a java program to

take an input array of integers and sort the elements in a descending order using selection sort.

```
import java.util.Arrays;
import java.util.Scanner;

public class SelectionsortDescending {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the size of the array: ");
        int len = sc.nextInt();
        int[] arr = new int[len];
        System.out.println("Enter array elements: ");
        for(int i=0;i<len;i++)
        {
            arr[i] = sc.nextInt();
        }
        System.out.print("Before sorting: ");
        System.out.println(Arrays.toString(arr));
        System.out.println();
        for(int i=0;i<len-1;i++)
        {
            int max=i;
            for(int j=i+1;j<len;j++)
            {
                if(arr[max]<arr[j])
                {
                    max=j;
                }
            }
            int temp = arr[max];
            arr[max] = arr[i];
            arr[i] = temp;
        }
        System.out.print("After sorting in Descending : ");
        System.out.println(Arrays.toString(arr));
        sc.close();
    }
}
```

Output:

11. Write a java program to

```
Enter the size of the array:
8
Enter array elements:
18
12
5
36
20
10
7
15
Before sorting: [18, 12, 5, 36, 20, 10, 7, 15]
After sorting in Descending : [36, 20, 18, 15, 12, 10, 7, 5]
```

achieve 100% abstraction.

```
interface Dog
{
    void bark();
    void jump();
}

class Puppy implements Dog
{
    public void bark() {
        System.out.println("Puppy is barking");
    }
    public void jump() {
        System.out.println("Puppy is jumping");
    }
}

public class AchieveAbstraction {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Dog obj = new Puppy();
        obj.bark();
        obj.jump();
    }
}
```

Output:

12. Write a java program to implement

```
Puppy is barking  
Puppy is jumping
```

method overloading.

13. Write a java program to

```
public class MethodOverloading {  
  
    static void exp()  
    {  
        System.out.println("Hello world");  
    }  
  
    static void exp(int a,double b)  
    {  
        double c = b/a;  
        System.out.println(c);  
    }  
    static void exp(String myText,int b)  
    {  
  
        System.out.println("Output will be: "+myText+" "+b);  
    }  
    static void exp(char a,char b,float c)  
    {  
        System.out.println(a+" "+b+" "+c);  
    }  
    static void exp(boolean myBool)  
    {  
  
        System.out.println(myBool+" "+myBool);  
    }  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        exp(false);  
        exp('B','Y',4);  
        exp(3,10);  
        exp("Hello",3);  
        exp();  
    }  
}
```

Output:

```
false false  
B Y 4.0  
3.3333333333333335  
Output will be: Hello 3  
Hello world
```



14. Write a java program to implement

method overriding.

```
class MethodOverRiding extends Sea {  
  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        MethodOverRiding obj = new MethodOverRiding();  
        //method overrides  
        obj.wave();  
        obj.flow();  
    }  
}  
  
class Sea {  
    void flow()  
    {  
        System.out.println("Sea is flowing ");  
    }  
    void wave()  
    {  
        System.out.println("Sea is waving ");  
    }  
}
```

Output:

```
Sea is waving  
Sea is flowing
```

Hybrid Inheritance.

15. Write a java program to implement

```
class HybridInheritance extends Cats {  
  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        HybridInheritance obj2 = new HybridInheritance();  
        obj2.running();  
        obj2.cry();  
        /* obj2.bark(); Compile Time Error:  
        the class HybridInheritance cannot access the methods of Class Dogs*/  
    }  
}  
  
class Animals  
{  
    void running()  
    {  
        System.out.println("All animals are running ");  
    }  
}  
  
class Dogs extends Animals  
{  
    void bark()  
    {  
        System.out.println("Dog barks!!");  
    }  
}  
  
class Cats extends Animals  
{  
    void cry()  
    {  
        System.out.println("Meow Meow!!!");  
    }  
}
```

Output:

```
All animals are running  
Meow Meow!!!
```

multilevel inheritance.

16. Write a java program to implement

```
class MultilevelInheritance extends Red { //child class of Red

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        MultilevelInheritance obj1 = new MultilevelInheritance();
        obj1.paint();
        obj1.spray();
    }

}

class Colour { //Parent Class
    void paint()
    {
        System.out.println("Colors are painted");
    }
}

class Red extends Colour { //child class of Colour
    void spray()
    {
        System.out.println("Colors are spraying");
    }
}
```

Output:

```
Colors are painted
Colors are spraying
```

17. Write a java program to

take input of integer array elements from the user and divide each element by the smallest element of the array and store the result in a resultant array. Implement Try- catch-finally block to counter null pointer divide by zero error.

```
import java.util.*;
public class TryCatchFinally {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        try {
            Scanner sc = new Scanner(System.in);
            System.out.println("Enter the array size");
            int size = sc.nextInt();
            int[] arr1 = new int[size];
            int[] arrcpy = new int[size];
            int[] res = new int[size];
            System.out.println("Enter the elements for array : ");
            for(int i=0;i<size;i++)
            {
                arr1[i] = sc.nextInt();
            }
            for(int i=0;i<size;i++) //copying the array elements:
            {
                arrcpy[i] = arr1[i];
            }
            int temp;
            for(int i=0;i<size;i++)
            {
                for(int j=i+1;j<size;j++)
                {
                    if(arr1[i]>arr1[j])
                    {
                        temp = arr1[i];
                        arr1[i] = arr1[j];
                        arr1[j] = temp;
                    }
                }
            }
            int small = arr1[0];

            for(int k=0;k<size;k++)
            {
                res[k] = arrcpy[k]/small;
            }
            System.out.println("Resultant array: ");
            System.out.println(Arrays.toString(res));
            sc.close();
        }
        catch(NullPointerException e)
        {
            System.out.println("Null Pointer Exception Caught");
        }
        catch(ArithmeticException c)
        {
            System.out.println("Divide by zero - Arithmetic exception");
        }
        finally
        {
            System.out.println("Finally block is executed");
        }
    }
}
```

Output:

18. Write a java program

```
Enter the array size
6
Enter the elements for array :
7
4
2
8
6
12
Resultant array:
[3, 2, 1, 4, 3, 6]
Finally block is executed
```

17. Write a java program to implement a constructor of the class, to print the instance variables value with respect to different objects.

```
class StudentClass {
    String name; //instance variables
    int age;
    public StudentClass(String name, int age)
    {
        this.name = name;
        this.age = age;
    }
    public String getName()
    {
        return name;
    }
    public int getAge()
    {
        return age;
    }
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        StudentClass s1 = new StudentClass("John",24);
        StudentClass s2 = new StudentClass("Ravi",50);
        System.out.println("The name of student 1: "+s1.getName());
        System.out.println("The age of student 2: "+s2.getAge());
    }
}
```

Output:

```
The name of student 1: John
The age of student 2: 50
```

to create a File at a particular location and to write to that particular file a String data which is taken as input from the user.

```

import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.util.Scanner;
public class FileCreateAndWrite {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        try {
            File f1 = new File("D:OutputFile.txt");
            FileWriter obj1 = new FileWriter(f1);
            Scanner sc = new Scanner(System.in);
            System.out.println("Enter the string: ");
            String str = sc.nextLine();
            obj1.write(str);
            sc.close();
            obj1.close();

        }
        catch(IOException e)
        {
            System.out.println("there is some issues");
            System.out.println(e);
        }
        System.out.println("file write done successfully");
    }
}

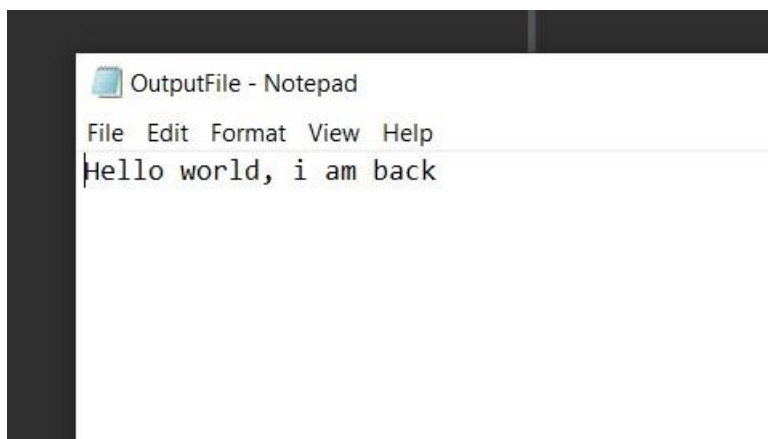
```

Output:

```

Enter the string:
Hello world, i am back
file write done successfully

```



20. Write a

program to read a file from a particular location and determine the number of vowels in that file.

```
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
public class FileReadVowels {
    public static void main(String[] args) throws IOException {
        // TODO Auto-generated method stub
        char[] array = new char[25];
        try {
            File f3 = new File("D:OutputFile.txt");
            FileReader fdr = new FileReader(f3);
            fdr.read(array);
            System.out.println("The string is: ");
            for(int j=0;j<array.length;j++)
            {
                System.out.print(array[j]);
            }

            int ctr = 0;
            for(int i=0;i<array.length;i++)
            {
                char ch = array[i];
                if(ch=='a' || ch=='e' || ch=='i' || ch=='o' || ch=='u' || ch=='A' || ch=='E' || ch=='I' || ch=='O' || ch=='U' )
                {
                    ctr++;
                }
            }
            System.out.println();
            System.out.println("Number of vowels is "+ctr);
            fdr.close();
        }
        catch(FileNotFoundException e)
        {
            System.out.println("there is some issue");
            System.out.println(e);
        }
    }
}
```

Output:

```
The string is:
Hello world, i am back
Number of vowels is 6
```

program to take input of two numbers from user. Now perform the particular arithmetic operation specified by the user and display the result.

Hint: use Switch case.



21.

the

```
import java.util.Scanner;

public class SwitchOperations {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter two number: ");
        double a=sc.nextInt();
        double b=sc.nextInt();

        System.out.println("1.Addition "+" 2.Subtraction "+" 3.Multiplication "+" 4.Division");
        System.out.println("Enter your choice ");
        int ch =sc.nextInt();
        switch(ch)
        {
            case 1:
                System.out.println("addition of two numbers: "+(a+b));
                break;
            case 2:
                System.out.println("Subtraction of two numbers: "+(a-b));
                break;
            case 3:
                System.out.println("Multiplication of two numbers: "+(a*b));
                break;
            case 4:
                System.out.println("Division of two numbers: "+(a/b));
                break;
            default:
                System.out.println("Invalid input");
        }
        sc.close();
    }
}
```

Output:

```
Enter two number:
64
16
1.Addition 2.Subtraction 3.Multiplication 4.Division
Enter your choice
4
Division of two numbers: 4.0
```

Create an array of 10 elements and print them using

for each loop.

22. Write a

the

```
public class ForEachLoop {  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
  
        String[] intArray = new String[]{"Apple", "Orange", "Banana", "Grapes", "Pappaya",  
            "JackFruit", "Kiwi", "Strawberry", "Mango", "Watermelon"};  
        System.out.println("Displaying array elements: ");  
  
        for(String item:intArray)  
        {  
            System.out.println(item+" ");  
        }  
    }  
}
```

Output:

```
Displaying array elements:  
Apple  
Orange  
Banana  
Grapes  
Pappaya  
JackFruit  
Kiwi  
Strawberry  
Mango  
Watermelon
```

23.

the

Take the number input from the console and add all positive numbers. (Not to consider the negative number if entered)

```
import java.util.Scanner;

public class SumofPositive {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        int sum=0;
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the limit : ");
        int len = sc.nextInt();
        int[] arr = new int[len];

        System.out.println("Enter the numbers : ");
        for(int i=0;i<len;i++)
        {
            arr[i] = sc.nextInt();
        }
        for(int i=0;i<len;i++)
        {
            if(arr[i]<0)
            {
                continue;
            }
            sum = sum+arr[i];
        }
        System.out.println("The sum of all positive numbers is: "+sum);
        sc.close();
    }
}
```

Output:

```
Enter the limit :
6
Enter the numbers :
7
-4
0
4
2
-10
The sum of all positive numbers is: 13
```

Create a labelled break and write a simple logic and execute the program.

```

public class LabelledBreak {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        int i=1;
        a:
        while(i<10)
        {
            if(i==5)
            {
                break a;// labeled break
            }
            System.out.println("i= "+i);
            i++;
        }
        System.out.println("Out of while loop");
    }

}

```

Output:

```

i= 1
i= 2
i= 3
i= 4
Out of while loop

```

24. Do the addition of around 10 even numbers, but use the continue statement in the logic.

```

public class EvenNumbersSum {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        int sum=0;
        System.out.println("first 10 even numbers: ");
        for(int i=1;i<=20;i++)
        {
            if(i%2!=0)
            {
                continue;
            }
            System.out.print(i+" ");
            sum = sum+i;
        }
        System.out.println();
        System.out.println("The sum of first 10 even numbers: "+sum);
    }

}

```

Output:

25.

```
first 10 even numbers:  
2 4 6 8 10 12 14 16 18 20  
The sum of first 10 even numbers: 110
```

25. Write a program to reverse the String (use char [] or String built in method)

```
public class StringReverse {  
  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        StringBuilder str = new StringBuilder("Hello world");  
        System.out.println("String : "+str.toString());  
        StringBuilder rev = str.reverse();  
        System.out.println("Reverse string: "+rev);  
    }  
}
```

Output:

```
String : Hello world  
Reverse string: dlrow olleH
```

26. Write programs to depict the usage of contains (), length (), replace (), concat (), equals ()

```
import java.util.Scanner;  
  
public class StringMethods {  
  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        Scanner sc = new Scanner(System.in);  
        System.out.println("Enter the string: ");  
        String st1 = sc.nextLine();  
        System.out.println("Enter another string");  
        String st2 = sc.nextLine();  
  
        System.out.println("Checking string 2 contains in string 1: "+st1.contains(st2));  
        System.out.println("The length of String1: "+st1.length());  
        System.out.println("Checking two strings are equal: "+st1.equals(st2));  
        System.out.println("Concatenating two string: "+st1.concat(st2));  
        System.out.println("Using replace method: "+st1.replace('l', 'k'));  
        sc.close();  
    }  
}
```



Output:

```
Enter the string:
Hello
Enter another string
ell
Checking string 2 contains in string 1:   true
The length of String1:   5
Checking two strings are equal:   false
Concatenating two string:   Helloell
Using replace method:   Hekko
```

27. Create an inheritance class. (Super class as Vehicle and 2 subclasses Car and Bike and inherit the Vehicle class methods)

```
public class InheritanceClass {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Bike obj1 = new Bike();
        Car obj2 = new Car();
        obj1.horn();
        obj1.Bikehorn();
        obj2.horn();
        obj2.Carhorn();
    }
}

class Vehicle
{
    void horn()
    {
        System.out.println("Horning");
    }
}

class Bike extends Vehicle
{
    void Bikehorn()
    {
        System.out.println("Bike horning");
    }
}

class Car extends Vehicle
{
    void Carhorn()
    {
        System.out.println("Car horning");
    }
}
```

```
Horning
Bike horning
Horning
Car horning
```



28.

Depict programmatically the Method overloading and Method overriding concepts.

```
public class Planets extends SolarSystem {  
  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        Planets obj1= new Planets();  
        obj1.rotate("Earth"); // method overriding  
        stop(); //method overloading  
        stop(5); //method overloading  
    }  
    static void stop()  
    {  
        System.out.println("Stops");  
    }  
  
    static void stop(int x)  
    {  
        System.out.println(x*2);  
    }  
}  
  
class SolarSystem  
{  
    void rotate(String name)  
    {  
        System.out.println(name +" is Rotating");  
    }  
}
```

Output:

```
Earth is Rotating  
Stops  
10
```

29.

Create an abstract class and extend that class and try to create the object of the abstract class in a program and execute.

```
abstract class Instruments
{
    abstract void sound(); //abstract method
    void string() //Non-abstract method
    {
        System.out.println("String instruments are good");
    }
}

public class AbstractExample {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Violin objs = new Violin();
        objs.sound();
        objs.string();
    }
}

class Violin extends Instruments
{
    void sound() //abstract method of Instrument class is implemented.
    {
        System.out.println("Abstract method overridden here");
    }
}
```

Output:

```
Abstract method overridden here
String instruments are good
```

30.

Write a java program to write the data into a file and read back using FileOutputStream/FileInputStream and also try the same using the BufferedReader and BufferedWriter

Using FileInputStream and FileOutputStream.

```
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;

public class FileReadAndWrite {

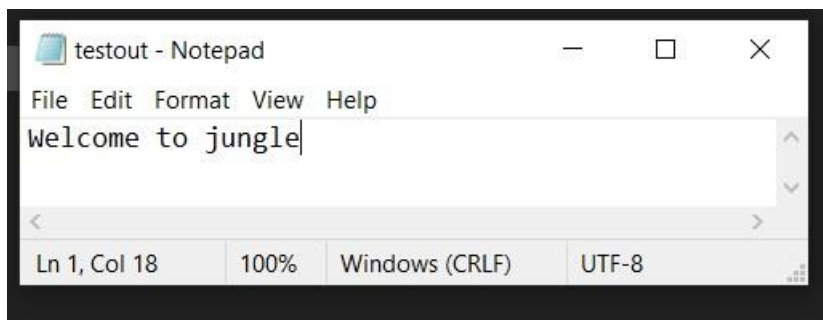
    public static void main(String[] args) throws IOException {
        // TODO Auto-generated method stub
        FileOutputStream fout = new FileOutputStream("D:testout.txt");
        String s="Welcome to jungle";
        byte b[]=s.getBytes();
        fout.write(b);
        fout.close();
        System.out.println("File created and wrote successfully ");
        System.out.println();
        FileInputStream fin = new FileInputStream("D:testout.txt");
        int i = 0;
        while((i=fin.read())!=-1)
        {
            System.out.print((char)i);
        }

        fin.close();
        System.out.println();
        System.out.println("File read successfully");
    }
}
```

```
File created and wrote successfully

File reading...

Welcome to jungle
```



## Using BufferedWriter and BufferedReader

```
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;

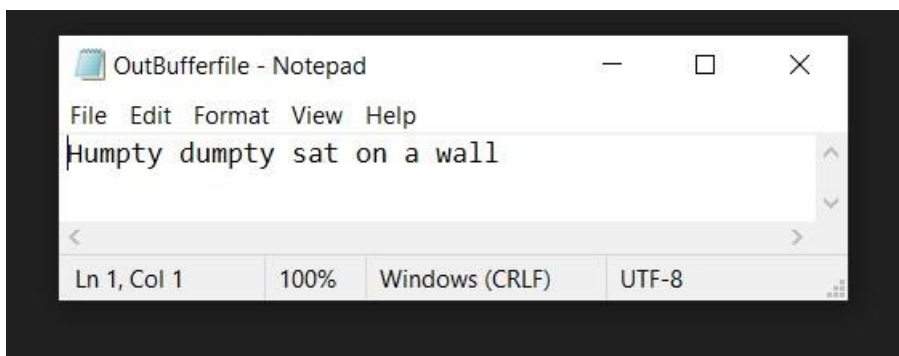
public class BufferReaderAndWriter {

    public static void main(String[] args) throws IOException {
        // TODO Auto-generated method stub
        FileWriter fw = new FileWriter("D:OutBufferfile.txt");
        BufferedWriter bw = new BufferedWriter(fw);
        bw.write("Humpty dumpty sat on a wall");
        bw.close();
        System.out.println("file write done");
        FileReader fr = new FileReader("D:OutBufferfile.txt");
        BufferedReader br = new BufferedReader(fr);
        System.out.println("file reading.... ");
        System.out.println();
        int i=0;
        while((i=br.read())!=-1)
        {
            System.out.print((char)i);
        }
        br.close();
    }
}
```

Output:

```
file write done
file reading....

Humpty dumpty sat on a wall
```



31. Write a java program to check the file owner details.

```
import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.nio.file.attribute.FileOwnerAttributeView;
import java.nio.file.attribute.UserPrincipal;

public class FileOwnerDetails {

    public static void main(String[] args) throws IOException {
        // TODO Auto-generated method stub
        Path path = Paths.get("D:OutputFile.txt");
        FileOwnerAttributeView ownerAttributeView = Files.getFileAttributeView(path, FileOwnerAttributeView.class);
        UserPrincipal owner = ownerAttributeView.getOwner();
        System.out.println("Owner name: "+owner.getName());
    }
}
```

Output:

```
Owner name: LAPTOP-C00R5473\USER
```

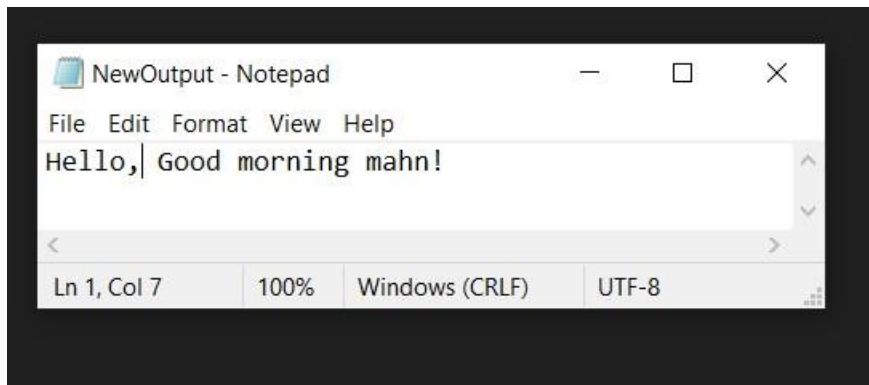
32. Write a java program to copy data from one file to another file.

```
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.util.Scanner;

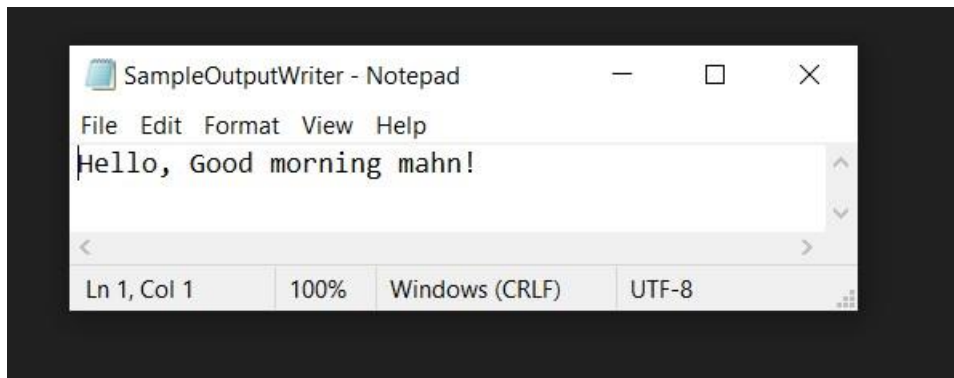
public class FileCopying {

    public static void main(String[] args) throws IOException {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter source file name ");
        String sfile = sc.next();
        System.out.println("Enter Destination file name ");
        String dfile = sc.next();
        FileReader fr = new FileReader(sfile);
        FileWriter fw = new FileWriter(dfile);
        int c;
        while((c = fr.read())!=-1)
        {
            fw.write(c);
        }
        System.out.println("copy finish");
        fr.close();
        fw.close();
        sc.close();
    }
}
```

Output:



```
Enter source file name
D:\NewOutput.txt
Enter Destination file name
D:\myfolder\SampleOutputWriter.txt
copy finish
```



33. How to add an element at a specific position in an ArrayList (create using <>)

```
import java.util.ArrayList;

public class ArrayListAdd {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        ArrayList<String> list = new ArrayList<>();
        list.add("Ram");
        list.add("Ravi");
        list.add("Anu");
        list.add(1,"John");
        list.add(0,"Qadi");

        for(String item: list)
        {
            System.out.print(item+" ");
        }
    }
}
```

Output:



34. Create an array of employee objects and iterate through it and remove the object at the 2nd position.

```
import java.util.Scanner;

public class ArrayOfObjectsDeletion {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter number of employees");
        int len = sc.nextInt();
        Employee[] arr = new Employee[len];
        System.out.println("Enter employee details: ");
        for(int i=0;i<len;i++)
        {
            System.out.println("Enter the employee "+(i+1)+" id: ");
            int empid = sc.nextInt();
            System.out.println("Enter the employee "+(i+1)+" Name: ");
            String empname = sc.next();
            arr[i] = new Employee(empid,empname);
        }
        System.out.println("Employee details before removing: ");
        for(int i=0;i<len;i++)
        {
            arr[i].display();
        }
        Employee[] newarr = new Employee[len-1];
        for(int i=0,k=0;i<len;i++)
        {
            if(i==1)
            {
                continue;
            }
            newarr[k++] = arr[i];
        }
        System.out.println("Removing second object: ");
        System.out.println();
        for(int i=0;i<len-1;i++)
        {
            newarr[i].display();
        }
        sc.close();
    }
}
```



```

class Employee
{
    public int id;
    public String name;

    Employee(int id,String name)
    {
        this.id = id;
        this.name = name;
    }
    public void display()
    {
        System.out.println(id + "-" + name);
    }
}

```

Output:

```

Enter number of employees
4
Enter employee details:
Enter the employee 1 id:
102
Enter the employee 1 Name:
Arun
Enter the employee 2 id:
104
Enter the employee 2 Name:
Nithya
Enter the employee 3 id:
108
Enter the employee 3 Name:
Kevin
Enter the employee 4 id:
121
Enter the employee 4 Name:
Tevez
Employee details before removing:
102-Arun
104-Nithya
108-Kevin
121-Tevez
Removing second object:

102-Arun
108-Kevin
121-Tevez

```

35. Create a HashMap type and display the elements using the keyset ()

```
import java.util.HashMap;

public class HashMapKeySet {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        HashMap<String,String> phonebook = new HashMap<String,String>();
        phonebook.put("Arun", "+917856431530");
        phonebook.put("Dsouza", "+41-205151");
        phonebook.put("Chun-li", "+5102014");
        System.out.println("Displaying PhoneBook: "+phonebook);

        System.out.println("The key set is: "+phonebook.keySet());
    }
}
```

Output:

```
Displaying PhoneBook: {Chun-li=+5102014, Arun=+917856431530, Dsouza=+41-205151}
The key set is: [Chun-li, Arun, Dsouza]
```