

Week 3

1. Write a Java program to calculate Sum & Average of an integer array.

```
import java.util.Scanner;

public class Sum_Avg_Array {

    public static void main(String args[]) {

        Scanner sc=new Scanner(System.in);
        int arr[]=new int[10],n=10;
        System.out.println("Enter 10 values");
        for(int i=0;i<n;i++) {
            arr[i]=sc.nextInt();
        }
        int sum=0, avg;

        for(int i=0;i<n;i++)H
        {
            sum+=arr[i];
        }
        avg=sum/n;
        System.out.println("Sum is "+sum);
        System.out.println("Avg is "+avg);
    }
}
```

```
}
Enter 10 values
1
2
3
4
5
6
7
8
9
10
Sum is 55
Avg is 5
```

2. Write a Java program to implement stack using array.

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

```

public class Stack {

    static final int MAX = 1000;
    static int top;
    int a[] = new int[MAX];

    boolean isEmpty()
    {
        return (top < 0);
    }
    Stack()
    {
        top = -1;
    }

    boolean push(int x)
    {
        if (top >= (MAX - 1)) {
            System.out.println("Stack Overflow");
            return false;
        }
        else {
            a[++top] = x;
            System.out.println(x + " pushed into stack");
            return true;
        }
    }

    int pop()
    {
        if (top < 0) {
            System.out.println("Stack Underflow");
            return 0;
        }
        else {
            int x = a[top--];
            return x;
        }
    }

    int peek()
    {
        if (top < 0) {
            System.out.println("Stack Underflow");
            return 0;
        }
        else {
            int x = a[top];
            return x;
        }
    }
}

```

```
}
```

```
public static void main(String args[])  
{  
    Stack s = new Stack();  
    s.push(10);  
    System.out.println( "Peek of the stack "+s.peek());  
    s.push(20);  
    System.out.println( "Peek of the stack "+s.peek());  
    s.push(30);  
    System.out.println( "Peek of the stack "+s.peek());  
    System.out.println(s.pop() + " Popped from stack");  
}
```

```
}
```

```
10 pushed into stack  
Peek of the stack 10  
20 pushed into stack  
Peek of the stack 20  
30 pushed into stack  
Peek of the stack 30  
30 Popped from stack
```

3. Write a Java program to implement Queue using array.

```
public class Queue {  
  
    int front, rear, size;  
    int capacity;  
    int array[];  
  
    public Queue(int capacity) {  
        this.capacity = capacity;  
        front = this.size = 0;  
        rear = capacity - 1;  
        array = new int[this.capacity];  
    }  
  
    boolean isFull(Queue queue)  
    { return (queue.size == queue.capacity);  
    }  
  
    boolean isEmpty(Queue queue)  
    { return (queue.size == 0); }  
  
    void enqueue( int item)
```

```

{
    if (isFull(this))
        return;
    this.rear = (this.rear + 1)%this.capacity;
    this.array[this.rear] = item;
    this.size = this.size + 1;
    System.out.println(item+ " enqueued to queue");
}

```

```

int dequeue()
{
    if (isEmpty(this))
        return Integer.MIN_VALUE;

    int item = this.array[this.front];
    this.front = (this.front + 1)%this.capacity;
    this.size = this.size - 1;
    return item;
}

```

```

int front()
{
    if (isEmpty(this))
        return Integer.MIN_VALUE;

    return this.array[this.front];
}

```

```

int rear()
{
    if (isEmpty(this))
        return Integer.MIN_VALUE;

    return this.array[this.rear];
}

```

```

public static void main(String[] args)
{
    Queue queue = new Queue(1000);

    queue.enqueue(10);
    queue.enqueue(20);
    queue.enqueue(30);
    queue.enqueue(40);

    System.out.println(queue.dequeue() +
        " dequeued from queue\n");

    System.out.println("Front item is " +

```

```

        queue.front());

        System.out.println("Rear item is " +
            queue.rear());
    }
}

```

10 enqueued to queue
 20 enqueued to queue
 30 enqueued to queue
 40 enqueued to queue
 10 dequeued from queue

Front item is 20
 Rear item is 40

4. Write a Java program to calculate Sum of two 2-dimensional arrays.

```

import java.util.Scanner;

public class Sum_2D_Array {

    public static void main(String args[]) {

        Scanner s = new Scanner(System.in);
        System.out.print("Enter number of rows: ");
        int rows = s.nextInt();
        System.out.print("Enter number of columns: ");
        int columns = s.nextInt();
        int[][] a = new int[rows][columns];
        int[][] b = new int[rows][columns];
        System.out.println("Enter the first matrix");
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < columns; j++) {
                a[i][j] = s.nextInt();
            }
        }
        System.out.println("Enter the second matrix");
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < columns; j++) {
                b[i][j] = s.nextInt();
            }
        }
        int[][] c = new int[rows][columns];
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < columns; j++) {
                c[i][j] = a[i][j] + b[i][j];
            }
        }
    }
}

```

```

        System.out.println("The sum of the two matrices is");
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < columns; j++) {
                System.out.print(c[i][j] + " ");
            }
            System.out.println();
        }
    }
}

```

Enter number of rows: 2

Enter number of columns: 2

Enter the first matrix

1

2

3

4

Enter the second matrix

1

2

3

4

The sum of the two matrices is

2 4

6 8

5. Write a Java program to find the range of a 1D array.

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

```
public class Array_Range {
```

```
    public static void main(String a[]){
```

```
        int[] arr = {2,4,2,4,5,6,3};
```

```
        System.out.println("My array elements:\n");
```

```
        for(int i=0;i<7;i++){
```

```
            System.out.print(arr[i]+" ");
```

```
        }
```

```
        int[] newArr = Arrays.copyOfRange(arr, 1, 4);
```

```
        System.out.println("\nMy new array elements:\n");
```

```
        for(int i=0;i<3;i++){
```

```
            System.out.print(newArr[i]+" ");
```

```
        }
```

```
    }
```

```
}
```

My array elements:

2 4 2 4 5 6 3

My new array elements:

4 2 4

6. Write a Java program to search an element in an array.

```
public class Linear_search {
    public static int search(int arr[], int x)
    {
        int n = arr.length;
        for(int i = 0; i < n; i++)
        {
            if(arr[i] == x)
                return i;
        }
        return -1;
    }

    public static void main(String args[])
    {
        int arr[] = { 2, 3, 4, 10, 40 };
        int x = 10;

        int n = arr.length, result=-1;
        for(int i = 0; i < n; i++)
        {
            if(arr[i] == x)
                result=i;
        }
        if(result == -1)
            System.out.print("Element is not present in array");
        else
            System.out.print("Element is present at index " + result);
    }
}
```

Element is present at index 3

7. Write a Java program to find the sum of even numbers in an integer array.

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

```
public class Sum_Even_Array {

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

```

Scanner sc=new Scanner(System.in);
int arr[]=new int[10],n=10;
System.out.println("Enter 10 values");
for(int i=0;i<n;i++)
{
    arr[i]=sc.nextInt();
}
int sum=0, avg;

for(int i=0;i<n;i++)
{
    if(arr[i]%2==0)
        sum+=arr[i];
}
System.out.println("Sum of even =" + sum);
}

}
Enter 10 values
1
2
3
4
5
6
7
8
9
10
Sum of even =30

```

8. Write a Java program to find the sum of diagonal elements in a 2D array.

```

public class Sum_of_Dia {

    static void printDiagonalSums(int [][]mat,
                                   int n)
    {
        int principal = 0, secondary = 0;
        for (int i = 0; i < n; i++) {
            principal += mat[i][i];
        }

        System.out.println(" Diagonal: "
                           + principal);
    }
}

```



```

static public void main (String[] args)
{
    int [][]a = { { 1, 2, 3, 4 },
                  { 5, 6, 7, 8 },
                  { 1, 2, 3, 4 },
                  { 5, 6, 7, 8 } };

    printDiagonalSums(a, 4);
}
}
Diagonal: 18

```

9. Reverse the elements in an array of integers without using a second array.

```

public class Reverse_An_Array {

    public static void main(String[] args)
    {
        int [] arr = {10, 20, 30, 40, 50};
        int n=5,i=0,j=n-1;
        while(i!=j)
        {
            int temp=arr[i];
            arr[i]=arr[j];
            arr[j]=temp;
            i++;
            j--;
        }
        System.out.println("Reversed array is: \n");
        for (int k = 0; k < n; k++) {
            System.out.println(arr[k]);
        }
    }
}

```

Output: Reversed array is:

```

50
40
30
20
10

```

10. Write a Java program to enter n elements in an array and find smallest number among them.

```

public class Reverse_An_Array {

    public static void main(String[] args)
    {
        int [] arr = {10, 20, -999, 40, 50};
        int n=5,min=arr[0];
        for (int k = 1; k < n; k++) {

            if(min>arr[k])
                min=arr[k];
        }

        System.out.println("Min element is "+min);
    }
}

```

Min element is -999

11. Write Java program to find the sum of all odd numbers in a 2D array.

```

public class Sum_of_ODD_Matrix {

    static void printDiagonalSums(int [][]mat,
                                   int n)
    {
        int sum=0;
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n; j++) {
                if(mat[i][j]%2!=0)
                    sum+=mat[i][j];
            }
        }

        System.out.println(" Sum of odd: "
                           + sum);
    }

    static public void main (String[] args)
    {
        int [][]a = { { 1, 2, 3, 4 },
                       { 5, 6, 7, 8 },
                       { 1, 2, 3, 4 },
                       { 5, 6, 7, 8 } };

        printDiagonalSums(a, 4);
    }
}

```

Sum of odd: 32

12. Write a Java program to print transpose of matrix.

```
public class Transpose {  
  
    static public void main (String[] args)  
    {  
        int [][]a = { { 1, 2, 3, 4 },  
                        { 5, 6, 7, 8 },  
                        { 1, 2, 3, 4 },  
                        { 5, 6, 7, 8 } };  
  
        int b[][]=new int[4][4],i,j;  
  
        for (i = 0; i < 4; i++)  
            for (j = 0; j < 4; j++)  
                b[i][j] = a[j][i];  
  
        System.out.print("Result matrix is \n");  
        for (i = 0; i < 4; i++)  
        {  
            for (j = 0; j < 4; j++)  
                System.out.print(b[i][j] + " ");  
            System.out.print("\n");  
        }  
    }  
}  
Result matrix is  
1 5 1 5  
2 6 2 6  
3 7 3 7  
4 8 4 8
```

13. Write a Java program to check whether a given matrix is sparse or not.

```
import java.util.Scanner;  
  
public class Reverse_An_Array {  
  
    public static void main(String args[])  
    {  
        Scanner sc = new Scanner(System.in);  
        System.out.println("Enter the dimensions of the matrix: ");  
        int m = sc.nextInt();
```

```

int n = sc.nextInt();
double[][] mat = new double[m][n];
int zeros = 0;
System.out.println("Enter the elements of the matrix: ");
for(int i=0; i<m; i++)
{
    for(int j=0; j<n; j++)
    {
        mat[i][j] = sc.nextDouble();
        if(mat[i][j] == 0)
        {
            zeros++;
        }
    }
}

if(zeros > (m*n)/2)
{
    System.out.println("The matrix is a sparse matrix");
}
else
{
    System.out.println("The matrix is not a sparse matrix");
}

sc.close();
}
}
Enter the dimensions of the matrix:
2
2
Enter the elements of the matrix:
0
0
0
1
The matrix is a sparse matrix

```

14. Write a Java program to count the prime numbers in an array.

```

public class Prime_Array {

    public static void main(String[] args) {
        int arr[]={1,2,3,4,5,6,7,8,9};

        for(int k=0;k<9;k++){

            int i,m=0,flag=0;

```

```

        int n=arr[k];
        m=n/2;
        if(n==0||n==1){
        }else{
            for(i=2;i<=m;i++){
                if(n%i==0){
                    flag=1;
                    break;
                }
            }
            if(flag==0) { System.out.println(n+" is prime number"); }
        }
    }
}
}
2 is prime number
3 is prime number
5 is prime number
7 is prime number

```

15. Write a Java program to find second highest element of an array.

```

public class Prime_Array {

    public static void main(String[] args) {
        int arr[]={1,2,3,4,5,6,7,8,9};

        int first,second;
        first = second = arr[0];
        for (int i = 0; i < 9; i++)
        {
            if (arr[i] > first)
            {
                second = first;
                first = arr[i];
            }

            else if (arr[i] > second && arr[i] != first)
                second = arr[i];
        }
        System.out.println("2nd is : "+second);
    }
}
2nd is : 8

```

16. Write a Java program which counts the non-zero elements in an integer array.

```

public class NonZero_Count {

    public static void main(String[] args) {
        int arr[]={1,2,3,0,0,0,0,8,9};

        int count=0;
        for (int i = 0; i < 9; i++)
        {
            if(arr[i]!=0)
                count++;
        }
        System.out.println("Non zero element : "+count);
    }
}

```

Non zero element : 5

17. Write a Java program to merge two float arrays.

```

public class Merge_Float {

    public static void main(String[] args) {
        float[]a = {1,2,3,4};
        float[]b = {4,16,1,2,3,22};
        float[]c = new float[a.length+b.length];
        int count = 0;

        for(int i = 0; i < a.length; i++) {
            c[i] = a[i];
            count++;
        }
        for(int j = 0; j < b.length;j++) {
            c[count++] = b[j];
        }
        for(int i = 0;i < c.length;i++) System.out.print(c[i]+" ");
    }
}

```

1.0 2.0 3.0 4.0 4.0 16.0 1.0 2.0 3.0 22.0

18. Write a Java program where elements of two integer arrays get added index wise and get stored into a third array.

```

public class Index_wise_add {

```

```

public static void main(String[] args) {
    int[]a = {1,2,3,4};
    int[]b = {4,16,1,2};
    int[]c = new int[4];
    int count = 0;

    for(int i = 0; i < a.length; i++) {
        c[i]=a[i]+b[i];
    }
    for(int i = 0;i < c.length;i++) System.out.print(c[i]+" ");
}

```

5 18 4 6

19. Write a Java program to multiply two matrices.

```

import java.util.Scanner;

public class Multiply_Matrix {

    public static void main(String[] args) {
        int m, n, p, q, sum = 0, c, d, k;

        Scanner in = new Scanner(System.in);
        System.out.println("Enter the number of rows and columns of first matrix");
        m = in.nextInt();
        n = in.nextInt();

        int first[][] = new int[m][n];

        System.out.println("Enter elements of first matrix");

        for (c = 0; c < m; c++)
            for (d = 0; d < n; d++)
                first[c][d] = in.nextInt();

        System.out.println("Enter the number of rows and columns of second matrix");
        p = in.nextInt();
        q = in.nextInt();

        if (n != p)
            System.out.println("The matrices can't be multiplied with each other.");
        else
        {
            int second[][] = new int[p][q];
            int multiply[][] = new int[m][q];

            System.out.println("Enter elements of second matrix");

```

```

    for (c = 0; c < p; c++)
        for (d = 0; d < q; d++)
            second[c][d] = in.nextInt();

    for (c = 0; c < m; c++)
    {
        for (d = 0; d < q; d++)
        {
            for (k = 0; k < p; k++)
            {
                sum = sum + first[c][k]*second[k][d];
            }

            multiply[c][d] = sum;
            sum = 0;
        }
    }

    System.out.println("Product of the matrices:");

    for (c = 0; c < m; c++)
    {
        for (d = 0; d < q; d++)
            System.out.print(multiply[c][d]+"\\t");

        System.out.print("\\n");
    }
}

```

Enter the number of rows and columns of first matrix

2

2

Enter elements of first matrix

10

20

30

40

Enter the number of rows and columns of second matrix

2

2

Enter elements of second matrix

40

30

20

10

Product of the matrices:

800 500

2000 1300

20. Write a Java program to subtract two matrices.

```
import java.util.Scanner;

public class Sub_Matrix {

    public static void main(String[] args) {
        int i, j;
        int mat1[][] = new int[3][3];
        int mat2[][] = new int[3][3];
        int mat3[][] = new int[3][3];
        Scanner scan = new Scanner(System.in);

        System.out.print("Enter Matrix 1 Elements : ");
        for(i=0; i<3; i++)
        {
            for(j=0; j<3; j++)
            {
                mat1[i][j] = scan.nextInt();
            }
        }

        System.out.print("Enter Matrix 2 Elements : ");
        for(i=0; i<3; i++)
        {
            for(j=0; j<3; j++)
            {
                mat2[i][j] = scan.nextInt();
            }
        }

        System.out.print("Subtracting Matrices (i.e. Matrix1 - Matrix2)...\n");
        for(i=0; i<3; i++)
        {
            for(j=0; j<3; j++)
            {
                mat3[i][j] = mat1[i][j] - mat2[i][j];
            }
        }

        System.out.print("Result of Matrix1 - Matrix2 is :\n");
        for(i=0; i<3; i++)
        {
            for(j=0; j<3; j++)
            {
                System.out.print(mat3[i][j]+ " ");
            }
            System.out.println();
        }
    }
}
```

```
    }  
}
```

Enter Matrix 1 Elements :

1
2
3
4
5
6
7
8
9

Enter Matrix 2 Elements : 9

8
7
6
5
4
3
2
1

Subtracting Matrices (i.e. Matrix1 - Matrix2)...

Result of Matrix1 - Matrix2 is :

-8 -6 -4
-2 0 2
4 6 8

21. Write a Java program to find duplicate elements in a 1D array and find their frequency of occurrence.

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

```
public class Remove_Duplicate {
```

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

```
        int arr[] = {1, 2, 2, 3, 4, 4, 4, 5, 5};  
        int n = arr.length;  
        int[] temp = new int[n];
```

```
        int j = 0;  
        for (int i=0; i<n-1; i++)
```

```
            if (arr[i] != arr[i+1])  
                temp[j++] = arr[i];
```

```
        temp[j++] = arr[n-1];
```

```

        for (int i=0; i<j; i++)
            arr[i] = temp[i];
        for (int i=0; i<j; i++)
            System.out.print(arr[i]+" ");
    }
}
1 2 3 4 5

```

22. Write a Java program to print every alternate number of a given array.

```

import java.util.Scanner;

public class Print_Alternative {

    public static void main (String[] args)
    {
        int arr[] = {1,2,3,4,5,6,7,8,9,10};
        int n = arr.length;

        for (int i=0; i<n; i+=2)
            System.out.print(arr[i]+" ");
    }
}
1 3 5 7 9

```

23. Given are two one-dimensional arrays A & B, which are sorted in ascending order. Write a Java program to merge them into single sorted array C that contains every item from arrays A & B, in ascending order.

```

public class Merge_Two_arrays {

    public static void main (String[] args)
    {
        int[] arr1 = {1, 3, 5, 7};
        int n1 = arr1.length;

        int[] arr2 = {2, 4, 6, 8};
        int n2 = arr2.length;

        int[] arr3 = new int[n1+n2];
        int i = 0, j = 0, k = 0;

        while (i<n1 && j <n2)
        {
            if (arr1[i] < arr2[j])

```

```

        arr3[k++] = arr1[i++];
    }
    else
        arr3[k++] = arr2[j++];
}

while (i < n1)
    arr3[k++] = arr1[i++];

while (j < n2)
    arr3[k++] = arr2[j++];

System.out.println("Array after merging");
for (i=0; i < n1+n2; i++)
    System.out.print(arr3[i] + " ");
}
}

```

Array after merging

1 2 3 4 5 6 7 8

24. Write a Java program to show 0-arguments constructor.

```

public class Constructor_Zero_Arg {

    public Constructor_Zero_Arg() {
        System.out.println("Hello");
    }

    public static void main(String[] args) {

        Constructor_Zero_Arg obj=new Constructor_Zero_Arg();

    }

}
Hello

```

25. Write a Java program to show parameterized constructor.

```

public class Constructor_Zero_Arg {

    public Constructor_Zero_Arg(String str) {
        System.out.println("Hello");
    }

    public static void main(String[] args) {

```

```

        String str="Hello";
        Constructor_Zero_Arg obj=new Constructor_Zero_Arg(str);
    }

}
Hello

```

26. Write a Java program to show constructor overloading.

```

public class Constructor_OverLoad {

    public Constructor_OverLoad() {
        System.out.println("UEMJ");
    }

    public Constructor_OverLoad(String str) {
        System.out.println(str);
    }

    public static void main(String[] args) {

        String str="UEMK";
        Constructor_OverLoad obj=new Constructor_OverLoad(str);

    }

}
UEMK

```

27. Write a class, Grader, which has an instance variable, score, an appropriate constructor and appropriate methods. A method, letterGrade() that returns the letter grade as O/E/A/B/C/F.

Now write a demo class to test the Grader class by reading a score from the user, using it to create a Grader object after validating that the value is not negative and is not greater than 100. Finally, call the letterGrade() method to get and print the grade.

```

import java.util.Scanner;

public class Constructor_Grade {

    public Constructor_Grade(float avg) {
        if(avg>=90)
        {
            System.out.print("A");
        }
    }
}

```

```

        else if(avg>=80 && avg<90)
        {
            System.out.print("B");
        }
        else if(avg>=60 && avg<80)
        {
            System.out.print("C");
        }
        else if(avg>=40 && avg<60)
        {
            System.out.print("D");
        }
        else
        {
            System.out.print("E");
        }
    }
}

```

```

public static void main(String[] args) {

```

```

    float avg;
    Scanner scanner = new Scanner(System.in);

    System.out.print("Enter the %: ");
    avg=scanner.nextFloat();
    System.out.print("The student Grade is: ");
    Constructor_Grade obj=new Constructor_Grade(avg);
}

```

```

}
Enter the %: 91
The student Grade is: A

```

28. Write a class, Commission, which has an instance variable, sales; an appropriate constructor; and a method, commission() that returns the commission.

Now write a demo class to test the Commission class by reading a sale from the user, using it to create a Commission object after validating that the value is not negative. Finally, call the commission() method to get and print the commission. If the sales are negative, your demo should print the message "Invalid Input".

```

import java.util.Scanner;

```

```

public class Commission {

```

```

    public Commission(int avg) {
        System.out.println("Commission is "+avg);
    }
}

```

```
public static void main(String[] args) {  
  
    int avg;  
    Scanner scanner = new Scanner(System.in);  
  
    System.out.print("Enter the value: ");  
    avg=scanner.nextInt();  
  
    Commission obj;  
        if(avg<0)  
            System.out.println("Invalid Input");  
    else  
        obj=new Commission(avg);  
    }  
}  
Enter the value: 999  
Commission is 999
```