

ASSIGNMENT-1

COURSE CODE- CS261

**COURSE NAME- OBJECT ORIENTED DESIGN AND PROGRAMMING
LABORATORY**

1. Find a mean and median of given 5 numbers.

```
import java.io.*;
import java.util.Scanner;
class mean_median
{
    public static void main(String args[])
    {
        int mean=0;
        Scanner sc=new Scanner(System.in);
        Integer arr[]=new Integer[5];
        System.out.println("Enter the numbers to find mean and median");
        for(int j=0;j<5;j++)
        {
            arr[j]=sc.nextInt();
        }
        for(int i=0;i<5;i++)
        {
            mean+=arr[i];
        }
        mean=mean/5;
        System.out.println("Mean of a given array is : "+mean);
        for(int i=0;i<4;i++)
        {
            for(int j=0;j<4-i;j++)
            {
                int temp=0;
                if(arr[j]>arr[j+1])
                {
                    temp=arr[j];
                    arr[j]=arr[j+1];
                    arr[j+1]=temp;
                }
            }
        }
    }
}
```

```
}  
}  
}
```

```
int mid=0;  
mid=5/2;  
int median=arr[mid];  
System.out.println("Median of a given numbers is : "+median);
```

```
}  
}
```

INPUT: *Enter the number to find mean and median*
 3 2 4 8 6

OUTPUT: *Mean of a given array is : 23*
 Median of a given numbers is : 4

2. Find an area and circumference of a circle.

```
import java.util.*;  
class circle  
{  
    public static void main(String args[])  
    {  
  
        float r,area,circum;  
        System.out.println("Enter the radius of circle");  
        Scanner sc=new Scanner(System.in);  
        r=sc.nextFloat();  
        area=(float)(3.14*r*r);  
        System.out.println("Area of circle: "+area);  
  
        circum=(float)(2*3.14*r);  
        System.out.println("circumference of circle: "+circum);
```

```
}  
}
```

INPUT: *Enter the radius of circle*

6

OUTPUT: *Area of circle: 113.04*
circumference of circle: 37.68

3. Find an area of a rectangle.

```
import java.util.Scanner;  
class area_rectangle  
{  
    public static void main(String args[])  
    {  
        double l,b,area;  
        Scanner sc=new Scanner(System.in);  
        System.out.println("Enter lenght of the rectangle");  
        l=sc.nextFloat();  
        System.out.println("Enter breadth of the rectangle");  
        b=sc.nextFloat();  
        area=l*b;  
        System.out.println("Area of rectangle is : "+area);  
    }  
}
```

INPUT: *Enter length of the rectangle*

5

Enter breadth of the rectangle

6

OUTPUT: *Area of rectangle is : 30*

4. Calculate the Simple Interest of a Bank Deposit.

```
import java.io.*;  
import java.util.Scanner;  
class simple_interest  
{  
    public static void main(String args[])  
    {  
        int y;  
        float s,p,r;  
        Scanner sc=new Scanner(System.in);
```

```

System.out.println("Enter the year");
y=sc.nextInt();
System.out.println("Enter the principle");
p=sc.nextFloat();
System.out.println("Enter the rate");
r=sc.nextFloat();
s=(p*r*y)/100;
System.out.println("Simple Interest is : ");
System.out.println(s);
}
}

```

INPUT: *Enter the year*
 2
 Enter the principle
 2000
 Enter the rate
 10%

OUTPUT: *Simple Interest is : 400.0*

5. Check if a given number is even or odd.

```

import java.util.*;
class even_odd
{
    public static void main(String args[])
    {
        int n;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the no.");
        n=sc.nextInt();
        if(n%2==0)
        {
            System.out.println("no. is even");
        }
        else
        {
            System.out.println("no. is odd");
        }
    }
}

```

INPUT: *Enter the no.*

6

OUTPUT: *no. is even*

INPUT: *Enter the no.*

3

OUTPUT: *no. is odd*

6. Check whether a number is Palindrome or Not.

```
import java.util.*;
class palindrome
{
    public static void main(String args[])
    {
        int n, rev=0, r;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the element");
        n=sc.nextInt();
        int temp=n;
        while(n>0)
        {
            r=n%10;
            n=(int)(n/10);
            rev=(rev*10)+r;
        }
        if(temp==rev)
        {
            System.out.println("No. is palindrome");
        }
        else
        {
            System.out.println("No. is not palindrome");
        }
    }
}
```

INPUT: *Enter the element*

252

OUTPUT: *No. is palindrome*

INPUT: *Enter the element*

372

OUTPUT: *No. is not palindrome*

7. Find largest number among 3 given numbers.

```
import java.util.*;
class largest
{
    public static void main(String args[])
    {
        int a,b,c;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the first element");
        a=sc.nextInt();
        System.out.println("Enter the Second element");
        b=sc.nextInt();
        System.out.println("Enter the third element");
        c=sc.nextInt();
        if(a>b&& a>c)
        {
            System.out.println("first is largest");
        }
        else if(b>a&& b>c)
        {
            System.out.println("second is largest");
        }
        else
        {
            System.out.println("third is largest");
        }
    }
}
```

INPUT: *Enter the first element*
5
Enter the second element
2
Enter the third element
1

OUTPUT: *first is largest*

8. Print first n elements of Fibonacci series.

```

import java.util.Scanner;
class fibonacci
{
    public static void main(String args[])
    {
        int n,a=0,b=1,temp=0;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the no.");
        n=sc.nextInt();
        System.out.println("Fibonacci series of a given no. is :");
        System.out.print(a+" "+b+" ");
        for(int i=3;i<=n;i++)
        {

            temp=b;
            b=a+b;
            a=temp;
            System.out.print(b+" ");
        }
    }
}

```

INPUT: *Enter the no.*

8

OUTPUT: *0 1 1 2 3 5 8 13*

9. Print table of any given number.

```

import java.util.Scanner;
class table
{
    public static void main(String args[])
    {
        int n,mul=0;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the no. to find the table");
        n=sc.nextInt();
        System.out.println("Table of a given no. is : ");
        for(int i=1;i<=10;i++)
        {
            mul=n*i;

```

```

        System.out.println(n+"*"+i+"="+mul);
    }
}

```

INPUT: *Enter the no. to find the table*

2

OUTPUT: *2*1=2*
*2*2=4*
*2*3=6*
*2*4=8*
*2*5=10*
*2*6=12*
*2*7=14*
*2*8=16*
*2*9=18*
*2*10=20*

10. Print factorial of a given number.

```

import java.util.Scanner;
class fact
{
    public static void main(String args[])
    {
        int n,fact=1;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the no.");
        n=sc.nextInt();
        for(int i=n;i>0;i--)
        {
            fact=fact*i;
        }
        System.out.println("Factorial of a given no. is : "+fact);
    }
}

```

INPUT: *Enter the no.*

5

OUTPUT: *Factorial of a given no. is : 120*

11. Check whether a number is a prime number or not.


```

import java.util.Scanner;
class prime_number
{
    public static void main(String args[])
    {
        int n;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the no.");
        n=sc.nextInt();
        int k=1;
        for(int i=2;i<=n/2;i++)
        {
            if(n%i==0)
            {
                k=0;
            }
        }
        if(k==1)
        {
            System.out.println("Number is a prime no.");
        }
        else
        {
            System.out.println("Number is not a prime no.");
        }
    }
}

```

INPUT: *Enter the no.*

5

OUTPUT: *Number is a prime no.*

INPUT: *Enter the no.*

9

OUTPUT: *Number is not a prime no.*

12. Print prime numbers between two given numbers.

```

import java.util.Scanner;
import java.lang.Math;

```

```

class prime
{
    public static void main(String args[])
    {
        int n,m,count=0,k=9;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the first no.");
        n=sc.nextInt();
        System.out.println("Enter the second no.");
        m=sc.nextInt();
        for(int i=n;i<=m;i++)
        {
            k=9;
            for(int j=2;j<=Math.sqrt(i);j++)
            {
                if((i%j)==0)
                {
                    k=0;
                }
            }
            System.out.println("Prime numbers between the given range is : ");
            if(k==9)
            {
                System.out.println(i);
            }
        }
    }
}

```

INPUT: *Enter the first no.*

2

Enter the second no.

10

OUTPUT: *Prime numbers between the given range is :*

2

3

5

7

13. Convert Fahrenheit into Centigrade.

```

import java.util.Scanner;
class temperature
{
    public static void main(String args[])
    {
        float f;
        float c;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the temperature in fahrenheit");
        f=sc.nextFloat();
        c=((float)(f-32)*5/9);
        System.out.println("Temperature in centigarde is : "+c);

    }
}

```

INPUT: *Enter the temperature in fahrenheit*
 4

OUTPUT: *Temperature in centigarde is : -15.555555*

14. Find the sum of digits of a given number.

```

import java.util.*;
class sod
{
    public static void main(String args[])
    {
        int n,remainder,sum=0;
        Scanner sc =new Scanner(System.in);
        System.out.println("Enter a number");
        n=sc.nextInt();
        int i=n;
        while(n!=0)
        {
            remainder=n%10;
            sum=sum+remainder;
            n=n/10;
        }
        System.out.println("Sum is : "+sum);

    }
}

```

INPUT: *Enter a number*

124

OUTPUT: *Sum is : 7*

15. Check whether a given number is an Armstrong number or not.

```
import java.util.*;
import java.lang.Math;
class arm
{
    public static void main(String args[])
    {
        int n;
        double r,sum=0;
        Scanner sc =new Scanner(System.in);
        System.out.println("Enter a number");
        n=sc.nextInt();
        int i=n;
        while(n!=0)
        {
            r=n%10;
            sum=sum+Math.pow(r,3);
            n=n/10;
        }
        if(sum==i)
        {
            System.out.println("Number is armstrong");
        }
        else
            System.out.println("Number is not armstrong");
    }
}
```

INPUT: *Enter a number*
371

OUTPUT: *Number is armstrong*

INPUT: *Enter a number*
543

OUTPUT: *Number is not armstrong*

16. Check whether a given year is a leap year or not.

```
import java.util.*;
class leap
{
    public static void main(String args[])
    {
        int n;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter A year");
        n=sc.nextInt();
        if(n%400==0)
        {
            System.out.println("It is a leap year");
        }
        else if(n%100==0)
        {
            System.out.println("It is not a leap year");
        }
        else if(n%4==0)
        {
            System.out.println("It is a leap year");
        }
        else
            System.out.println("It is not a leap year");
    }
}
```

INPUT: *Enter A year*
 2016

OUTPUT: *It is a leap year*

INPUT: *Enter A year*
 2015

OUTPUT: *It is not a leap year*

INPUT: *Enter A year*
 3000

OUTPUT: *It is not a leap year*

17. Implement binary search to search a given element.

```
import java.util.Scanner;
class binary_search
{
    public static void main(String args[])
    {
        int lb,ub,mid=0,n,flag=0;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the no.");
        n=sc.nextInt();
        lb=0;
        ub=n-1;
        Integer arr[]=new Integer[n];
        System.out.println("Enter the no. of element in sorted order");
        for(int i=0;i<n;i++)
        {
            arr[i]=sc.nextInt();
        }
        System.out.println("Enter the no.to be search");
        int key=sc.nextInt();
        while(lb<=ub)
        {
            mid=(lb+ub)/2;
            if(arr[mid]>key)
            {
                ub=mid-1;
            }
            else if(arr[mid]<key)
            {
                lb=mid+1;
            }
            else
            {
                flag=1;
                key=arr[mid];
                break;
            }
        }
        if(flag==1)
        {
            System.out.println("Element found");
        }
        else
        {
            System.out.println("Element not found");
        }
    }
}
```

```

        System.out.println("No. found at index: "+ mid);
    }
    else
    {
        System.out.println("No. not found in the given array");
    }
}
}

```

INPUT: *Enter the no.*

5

Enter the no. of element in sorted order

4 7 9 50 61

Enter the no.to be search

50

OUTPUT: *No. found at index: 3*

INPUT: *Enter the no.*

8

Enter the no. of element in sorted order

3 9 87 98 100 107 234 290

Enter the no.to be search

342

OUTPUT: *No. not found in the given array*

18. Implement insertion sort to sort a given array.

```

import java.util.Scanner;
class insertion_sort
{
    public static void main(String args[])
    {
        int n;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the size of array");
        n=sc.nextInt();
        Integer arr[]=new Integer[n];
        System.out.println("Enter element of array");
        for(int i=0;i<n;i++)
        {
            arr[i]=sc.nextInt();

```

```

    }
    for(int i=1;i<n;i++)
    {
        int key=arr[i];
        int j=i-1;
        while(j>=0&&arr[j]>key)
        {
arr[j+1]=arr[j];
j--;
        }
        arr[j+1]=key;
    }
    System.out.print("Sorted array is: ");
    for(int j=0;j<n;j++)
    {
        System.out.print(arr[j]+" ");
    }
}

```

INPUT: *Enter a no.*

5

Enter no.of element

4 8 4 10 1

OUTPUT: *Sorted array is:*

1 4 4 8 10

19. Implement quicksort to sort a given array.

```

import java.util.Scanner;
class quick
{
    int partition(int arr[],int low,int high)
    {
        int pivot=arr[high];
        int i=low-1;
        for(int j=low;j<high;j++)
        {
            if(arr[j]<=pivot)
            {
                i++;
            }
        }
    }
}

```



```

        int temp=arr[i];
        arr[i]=arr[j];
        arr[j]=temp;

    }
}
int temp=arr[i+1];
arr[i+1]=arr[high];
arr[high]=temp;
return i+1;
}

void sort(int arr[],int low,int high)
{
    if(low<high)
    {
        int pi=partition(arr,low,high);
        sort(arr,low,pi-1);
        sort(arr,pi+1,high);
    }
}

static void printArray(int arr[],int n)
{
    for(int i=0;i<n;i++)
    {
        System.out.print(arr[i]+" ");
    }
}

public static void main(String args[])
{
    int n;

    Scanner sc=new Scanner(System.in);
    System.out.println("Enter the size of array");
    n=sc.nextInt();
    int arr[]=new int[n];
    System.out.println("Enter the elements of array");
    for(int i=0;i<n;i++)
    {

```

```

        arr[i]=sc.nextInt();
    }
    quick ob=new quick();
    ob.sort(arr,0,n-1);
    System.out.println("Sorted Array: ");
    ob.printArray(arr,n);

}

}

```

INPUT: *Enter the size of array*
 8
 Enter the elements of array
 7 4 2 9 10 33 1 5

OUTPUT: *Sorted Array:*
 1 2 4 5 7 9 10 33

20. Given an input sequence, implement a binary search tree.

```

import java.util.*;
class BST1
{
    class Node
    {
        Integer val;
        Node left,right;
        Node(int val)
        {
            left=right=null;
            this.val=val;
        }
    }
    Node root;
    BST1()
    {
        root=null;
    }
    public void insert(int val)

```

```

{
    Node ob1=new Node(val);
    if(root==null)
        root=ob1;
else{
    Node hop=root;
    while(true)
    {
        if(val>hop.val)
        {
            if(hop.right==null)
                break;
            else
            {
                hop=hop.right;}
        }
        else
        {
            if(hop.left==null)
                break;
            else
                hop=hop.left;
        }
    }
    if(val>hop.val)
    {
        hop.right=ob1;
    }
    else
        hop.left=ob1;
    }
}

public void traverse()
{
    System.out.print("inorder:");
    inorder(root);
    System.out.println();
    System.out.print("preorder:");
    preorder(root);
    System.out.println();
    System.out.print("postorder:");

```

```

        postorder(root);
        System.out.println();
    }
    void inorder(Node root)
    {

        if(root!=null)
        {
            inorder(root.left);
            System.out.print(root.val+" ");
            inorder(root.right);
        }
    }

    void preorder(Node root)
    {

        if(root!=null)
        {
            System.out.print(root.val+" ");
            preorder(root.left);
            preorder(root.right);
        }
    }
    void postorder(Node root)
    {

        if(root!=null)
        {

            postorder(root.left);
            postorder(root.right);
            System.out.print(root.val+" ");
        }
    }
}

```

// DELETION IN BST//

```

void DELETE_OB(int val)
{

```

```

        Node ob= SEARCH(val);
        Node pob=parent(val);
        if(ob.right==null&&ob.left==null)
            DELETE_leaf(ob,pob);
        if(ob.right==null || ob.left==null)
            DELETE_single(ob,pob);
    }
    void DELETE_leaf( Node ob,Node pob)
    {

        if(pob.right==ob)
        {
            pob.right=null;
        }
        else
        {pob.left=null;}
    }

    Node SEARCH(int val)

    {
Node current=root;
        while(current.val!=val)
        {
            if(current.val<=val)
                current=current.right;
            else
                current=current.left;
        }
        return current;
    }
    Node parent(int val)
    {
Node hop=root;
        Node current=root;
        while(current.val!=val)

```

```

        {
            hop=current;
            if(current.val<=val)
                current=current.right;
            else
                current=current.left;
        }
        return hop;
    }

    void DELETE_single(Node ob,Node pob)
    {

        if(pob.right==ob)
        {
            if(ob.right!=null)
                pob.right=ob.right;
            if(ob.left!=null)
                pob.right=ob.left;
        }
        if(pob.left==ob)
        {
            if(ob.right!=null)
            {
                pob.left=ob.right;
            }
            if(ob.left!=null)
            {
                pob.left=ob.left;
            }
        }
    }
}

```

```

public static void main(String args[])
{
    BST1 b=new BST1();
    int n;

    System.out.println("Enter the size of array");
    Scanner sc=new Scanner(System.in);
    n=sc.nextInt();
    Integer arr[]=new Integer[n];
}

```

```

        System.out.println("Enter the element");
        for(int i=0;i<n;i++)
        {
arr[i]=sc.nextInt();
b.insert(arr[i]);
        }

        b.traverse();
        System.out.println("Enter element to delete");
        int d=sc.nextInt();
        b.DELETE_OB(d);

        b.traverse();

    }
}

```

INPUT:

```

Enter the size of array
5
Enter the element
38
90
42
1
3

```

OUTPUT:

```

Traversing before deleting a element
Inorder traversal:1 3 38 42 90
Preorder traversal:38 1 3 90 42
Postorder traversal:3 1 42 90 38
Enter the element to delete
42
Traversing after deleting a element
Inorder traversal:1 3 38 90
Preorder traversal:38 1 3 90
Postorder traversal:3 1 90 38

```

NAME-AMAN KUMAR KANOJIA
ROLL NO.-201851014
SECTION-1
GROUP-1A