

Experiment No: 1**Date:****Aim: 1) Basic Operations in Java Programming****1.1) Write a Java program to display default value of all primitive data type of JAVA.****Source Code:**

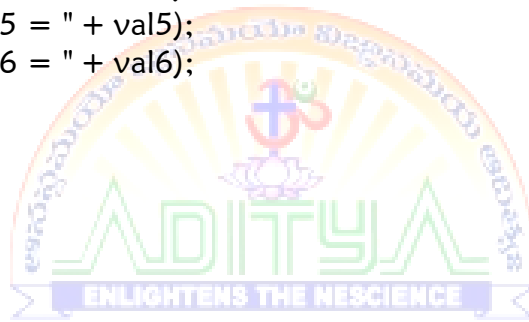
```

public class Demo
{
    static boolean val1;
    static double val2;
    static float val3;
    static int val4;
    static long val5;
    static String val6;
    public static void main(String[] args)
    {
        System.out.println("Default values.....");
        System.out.println("Val1 = " + val1);
        System.out.println("Val2 = " + val2);
        System.out.println("Val3 = " + val3);
        System.out.println("Val4 = " + val4);
        System.out.println("Val5 = " + val5);
        System.out.println("Val6 = " + val6);
    }
}

```

Output:

Default values.....
 Val1 = false
 Val2 = 0.0
 Val3 = 0.0
 Val4 = 0
 Val5 = 0
 Val6 = null

**Regd No:**

--	--	--	--	--	--	--	--	--	--

1.2) Write a Java program to find the discriminant value D and find out the roots of the quadratic equation of the form $ax^2+bx+c=0$.

Source Code:

```
import java.util.*;
public class Roots
{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        // value a, b, and c
        double a , b, c;
        double root1, root2;

        System.out.println("Enter a, b and c values");
        a=sc.nextDouble();
        b=sc.nextDouble();
        c=sc.nextDouble();

        // calculate the determinant (b2 - 4ac)
        double determinant = b * b - 4 * a * c;

        // check if determinant is greater than 0
        if (determinant > 0) {

            // two real and distinct roots
            root1 = (-b + Math.sqrt(determinant)) / (2 * a);
            root2 = (-b - Math.sqrt(determinant)) / (2 * a);

            System.out.format("root1 = %.2f and root2 = %.2f", root1, root2);
        }

        // check if determinant is equal to 0
        else if (determinant == 0) {

            // roots are equal
            root1 = root2 = -b / (2 * a);
            System.out.format("root1 = root2 = %.2f;", root1);
        }

        // if determinant is less than zero
        else {

            // roots are complex number and distinct
            double real = -b / (2 * a);
            double imaginary = Math.sqrt(-determinant) / (2 * a);
            System.out.format("root1 = %.2f+%.2fi", real, imaginary);
            System.out.format("\nroot2 = %.2f-%.2fi", real, imaginary);
        }
    }
}
```

Regd No:

--	--	--	--	--	--	--	--	--	--

Output:

D:\>javac Roots.java

D:\>java Roots

Enter a, b and c values

2

3

4

root1 = -0.75+1.20i

root2 = -0.75-1.20i



Regd No:

--	--	--	--	--	--	--	--	--	--

- 1.3) Five Bikers Compete in a race such that they drive at a constant speed which may or may not be the same as the other. To qualify the race, the speed of a racer must be more than the average speed of all 5 racers. Take as input the speed of each racer and print back the speed of qualifying racers.

Source Code:

```
import java.util.Scanner;
class Bike_Racers
{
    public static void main(String args[])
    {
        Scanner sc=new Scanner(System.in);
        int racer1_Speed,racer2_Speed,racer3_Speed,racer4_Speed,racer5_Speed;
        int sum;
        float avg_Speed;

        System.out.println("Enter 5 Bike Racers Speeds");
        racer1_Speed=sc.nextInt();
        racer2_Speed=sc.nextInt();
        racer3_Speed=sc.nextInt();
        racer4_Speed=sc.nextInt();
        racer5_Speed=sc.nextInt();

        sum=racer1_Speed+racer2_Speed+racer3_Speed+racer4_Speed+racer5_Speed;
        avg_Speed=(float)sum/5;

        System.out.println("Average Speed is:"+avg_Speed);

        System.out.println("The Qualified Racers are:");
        if(racer1_Speed>avg_Speed)
            System.out.println(racer1_Speed);
        if(racer2_Speed>avg_Speed)
            System.out.println(racer2_Speed);
        if(racer3_Speed>avg_Speed)
            System.out.println(racer3_Speed);
        if(racer4_Speed>avg_Speed)
            System.out.println(racer4_Speed);
        if(racer5_Speed>avg_Speed)
            System.out.println(racer5_Speed);
    }
}
```

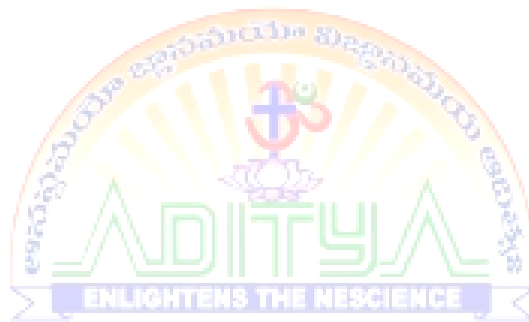
Regd No:

--	--	--	--	--	--	--	--	--	--

Output:

```
D:\Java_Programs>javac Bike_Racers.java

D:\Java_Programs>java Bike_Racers
Enter 5 Bike Racers Speeds
255
247
289
220
236
Average Speed is:249.4
The Qualified Racers are:
255
289
```

**Regd No:**

--	--	--	--	--	--	--	--	--	--

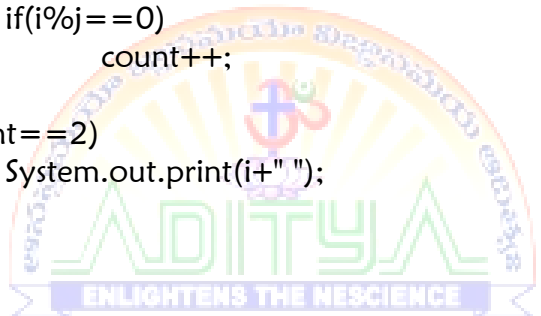
Experiment No: 2**Date:****Aim: 2) Control Flow Statements****2.1) Write a Java program to select all the prime numbers within the range of 1to100.****Source Code:**

```

import java.util.Scanner;
class Prime_Numbers
{
    public static void main(String args[])
    {
        Scanner sc=new Scanner(System.in);
        int num,i,j,count=0;
        System.out.println("Enter a number to find prime numbers upto it");
        num=sc.nextInt();

        for(i=2;i<=num;i++)
        {
            count=0;
            for(j=1;j<=i;j++)
            {
                if(i%j==0)
                    count++;
            }
            if(count==2)
                System.out.print(i+" ");
        }
    }
}

```


Output:

D:\Java_Programs>java Prime_Numbers
Enter a number to find prime numbers upto it
100

2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97

Regd No:

--	--	--	--	--	--	--	--	--	--

2.2) Write a Java program to Find the sum of all even terms in the Fibonacci sequence up to the given range N.

Source Code:

```
import java.util.Scanner;
class Fibbonaci
{
    public static void main(String args[])
    {
        int n1=0,n2=1,n3,n,sum=0;
        n3=n1+n2;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the n value");
        n=sc.nextInt();

        while(n3<=n)
        {
            if(n3%2==0)
                sum=sum+n3;
            n1=n2;
            n2=n3;
            n3=n1+n2;
        }
        System.out.println("sum of all even terms in the Fibonacci sequence up to the given
        range "+n+" is: "+sum);
    }
}
```

Output:

D:\Java_Programs>java Fibbonaci

Enter the n value

15

sum of all even terms in the Fibonacci sequence up to the given range 15 is: 10

D:\Java_Programs>java Fibbonaci

Enter the n value

150

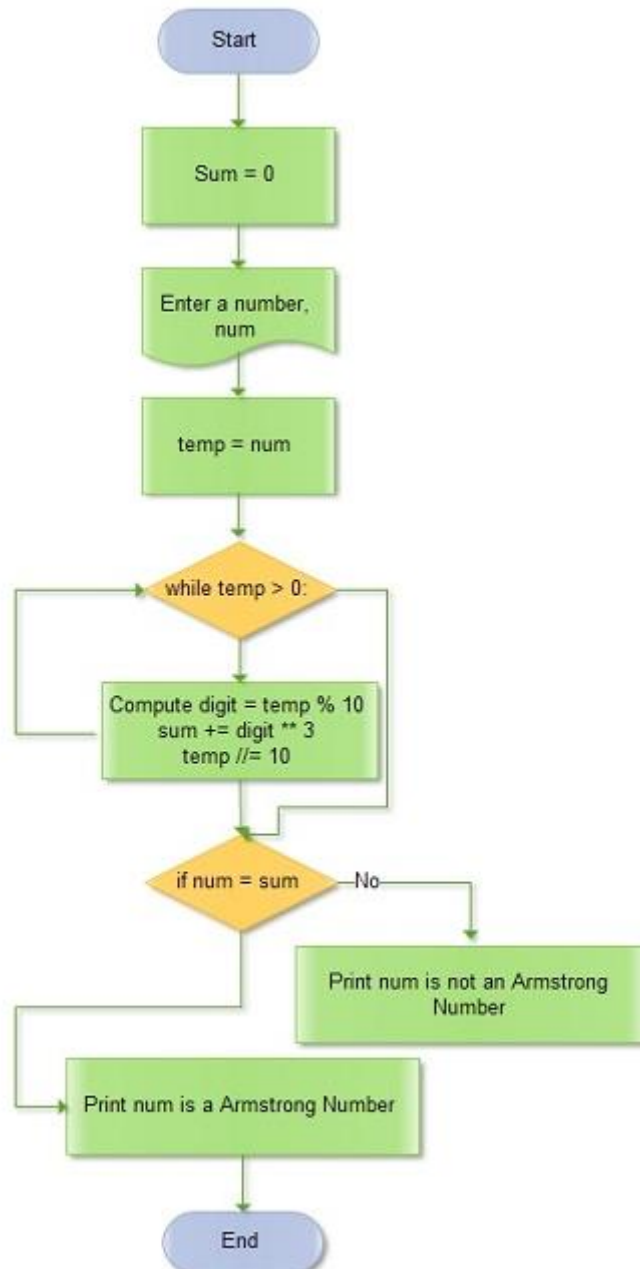
sum of all even terms in the Fibonacci sequence up to the given range 150 is: 188

Regd No:

--	--	--	--	--	--	--	--	--	--

2.3) Write a Java program to check whether a given number is Armstrong or not.

Definition: An Armstrong number or Narcissistic number is an n-digit number equivalent to the sum of digits raised to the nth power of digits from the number. A few Armstrong numbers are: 0, 1, 2, 3, 153, 370, 407, 1634, 8208, etc.



Flow chart to find the given 3 digit number is Armstrong or not

Source Code:

```

import java.util.Scanner;
class Armstrong
{
    public static void main(String args[])
    {
        Scanner sc=new Scanner(System.in);
        int num,sum=0,rem,m,d;
        System.out.println("Enter a number");
        num=sc.nextInt();

        d=(int)Math.log10(num)+1;
        m=num;
        while(num>0)
        {
            rem=num%10;
            sum=sum+(int)Math.pow(rem,d);
            num=num/10;
        }

        if(sum==m)
            System.out.println(m+" is Armstrong Number");
        else
            System.out.println(m+" is not an Armstrong Number");
    }
}

```

**Output:**

```

D:\Java_Programs>java Armstrong
Enter a number
1634
1634 is Armstrong Number

```

```

D:\Java_Programs>java Armstrong
Enter a number
125
125 is not an Armstrong Number

```

Regd No:

--	--	--	--	--	--	--	--	--	--

Experiment No: 3**Date:****Aim: 3) Working with Arrays****3.1) Write a Java program to implement binary search.****Source Code:**

```

import java.util.*;
class BinarySearchExample
{
    public static void binarySearch(int arr[], int first, int last, int key)
    {
        int mid = (first + last)/2;
        while( first <= last )
        {
            if ( arr[mid] < key )
            {
                first = mid + 1;
            }
            else if(arr[mid] == key )
            {
                System.out.println("Element is found at index: " + mid);
                break;
            }
            else
            {
                last = mid - 1;
            }
            mid = (first + last)/2;
        }
        if ( first > last )
        {
            System.out.println("Element is not found!");
        }
    }
    public static void main(String args[])
    {
        Scanner sc=new Scanner(System.in);
        int n,key,arr[];
        System.out.println("Enter the number of elements");
        n=sc.nextInt();
        arr=new int[n];
        System.out.println("Enter "+n+" elements");
        for(int i=0;i<n;i++)
            arr[i]=sc.nextInt();
        System.out.println("Enter the number to search");
        key=sc.nextInt();
        int last=n-1;
        binarySearch(arr,0,last,key);
    }
}

```

Regd No:

--	--	--	--	--	--	--	--	--	--

Output:

```
D:\>java BinarySearchExample
Enter the number of elements
8
Enter 8 elements
11 22 33 44 55 66 77 88
Enter the number to search
44
Element is found at index: 3
```

**Regd No:**

--	--	--	--	--	--	--	--	--	--

3.2) Write a Java program to sort for an element in a given list of elements using bubble sort.

Source Code:

```
import java.util.Scanner;
class BubbleSortExample
{
    public static void bubbleSort(int[] arr)
    {
        int n=arr.length,temp;
        for(int i=0;i<n-1;i++)
        {
            for(int j=0;j<n-i-1;j++)
            {
                if(arr[j]>arr[j+1])
                {
                    temp=arr[j];
                    arr[j]=arr[j+1];
                    arr[j+1]=temp;
                }
            }
        }
    }
    public static void main(String[] args)
    {
        int arr[],n;
        Scanner sc=new Scanner(System.in);
        n=sc.nextInt();
        arr=new int[n];

        for(int i=0;i<n;i++)
            arr[i]=sc.nextInt();

        System.out.println("Array Before Bubble Sort");
        for(int i=0; i < arr.length; i++)
        {
            System.out.print(arr[i] + " ");
        }
        System.out.println();

        bubbleSort(arr);        //sorting array elements using bubble sort

        System.out.println("Array After Bubble Sort");
        for(int i=0; i < arr.length; i++)
        {
            System.out.print(arr[i] + " ");
        }
    }
}
```



Output:

D:\Practice>java BubbleSortExample

8

1 4 2 3 5 9 8 6

Array Before Bubble Sort

1 4 2 3 5 9 8 6

Array After Bubble Sort

1 2 3 4 5 6 8 9



Regd No:

--	--	--	--	--	--	--	--	--	--

3.3) Write a Java program to sort for an element in a given list of elements using merge sort.**Source Code:**

```

class Merge_Sort
{
    /* Function to merge the subarrays of a[] */
    void merge(int a[], int beg, int mid, int end)
    {
        int i, j, k;
        int n1 = mid - beg + 1;
        int n2 = end - mid;

        /* temporary Arrays */
        int LeftArray[] = new int[n1];
        int RightArray[] = new int[n2];

        /* copy data to temp arrays */
        for (i = 0; i < n1; i++)
            LeftArray[i] = a[beg + i];
        for (j = 0; j < n2; j++)
            RightArray[j] = a[mid + 1 + j];

        i = 0;      /* initial index of first sub-array */
        j = 0;      /* initial index of second sub-array */
        k = beg;    /* initial index of merged sub-array */

        while (i < n1 && j < n2)
        {
            if(LeftArray[i] <= RightArray[j])
            {
                a[k] = LeftArray[i];
                i++;
            }
            else
            {
                a[k] = RightArray[j];
                j++;
            }
            k++;
        }
        while (i < n1)
        {
            a[k] = LeftArray[i];
            i++;
        }
    }
}

```

Regd No:

--	--	--	--	--	--	--	--	--	--

```

        k++;
    }

    while(j<n2)
    {
        a[k]=RightArray[j];
        j++;
        k++;
    }
}

void mergeSort(int a[], int beg, int end)
{
    if(beg < end)
    {
        int mid = (beg + end) / 2;
        mergeSort(a, beg, mid);
        mergeSort(a, mid + 1, end);
        merge(a, beg, mid, end);
    }
}

/* Function to print the array */
void printArray(int a[], int n)
{
    int i;
    for (i = 0; i < n; i++)
        System.out.print(a[i] + " ");
}

public static void main(String args[])
{
    int a[] = { 11, 30, 24, 7, 31, 16, 39, 41 };
    int n = a.length;
    Merge_Sort m1 = new Merge_Sort();
    System.out.println("\nBefore sorting array elements are - ");
    m1.printArray(a, n);
    m1.mergeSort(a, 0, n - 1);
    System.out.println("\nAfter sorting array elements are - ");
    m1.printArray(a, n);
    System.out.println("");
}
}

```

Regd No:

--	--	--	--	--	--	--	--	--	--

Output:

D:\JAVA-Programs\Arrays>java Merge_Sort

Before sorting array elements are -

11 30 24 7 31 16 39 41

After sorting array elements are -

7 11 16 24 30 31 39 41



Regd No:

--	--	--	--	--	--	--	--	--	--

Experiment No: 4**Date:****Aim: 4) Class Mechanism**

4.1) Write a Java program to display the details of a person. Personal details should be given in one method and the qualification details in another method.

Source Code:

```
import java.util.*;
class Employee
{
    int empid;
    String empname,desg, Organization,ug,pg,address;
    float Sal,per_marks_ug,per_marks_pg;

    Scanner sc=new Scanner(System.in);

    public void get_PersonalInfo()
    {
        System.out.println("Enter your empid, name, desg, salary, organization name, address");
        empid=sc.nextInt();
        sc.nextLine();
        empname=sc.nextLine();
        desg=sc.nextLine();
        Sal=sc.nextFloat();
        sc.nextLine();
        Organization=sc.nextLine();
        address=sc.nextLine();
    }
    public void get_QualificationInfo()
    {
        System.out.println("Enter your UG course, marks and PG course and Marks");
        ug=sc.nextLine();
        per_marks_ug=sc.nextFloat();
        sc.nextLine();
        pg=sc.nextLine();
        per_marks_pg=sc.nextFloat();
    }
    public void show_PersonalInfo()
    {
        System.out.println("=====");
        System.out.println("                PERSONAL INFORMATION                ");
        System.out.println("=====");
        System.out.println("EMPID: "+empid);
        System.out.println("EMP NAME: "+empname);
        System.out.println("DESGINATION: "+desg);
        System.out.println("SALARY: "+Sal);
        System.out.println("ORGANIZATION NAME: "+Organization);
        System.out.println("ADDRESS: "+address);
    }
}
```

Regd No:

--	--	--	--	--	--	--	--	--	--

```

public void show_QualificationInfo()
{
    System.out.println("=====");
    System.out.println("                QUALIFICATION INFORMATION                ");
    System.out.println("=====");

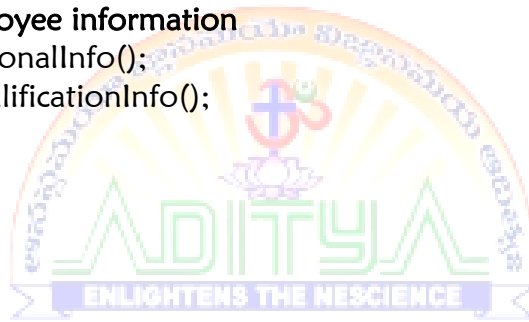
    System.out.println("UG COURSE: "+ug);
    System.out.println("UG PERCENTAGE: "+per_marks_ug);
    System.out.println("PG COURSE: "+pg);
    System.out.println("PG PERCENTAGE: "+per_marks_pg);

}

public static void main(String args[])
{
    // Object Creation - memory for member variable declared inside the class
    Employee e1=new Employee();
    // reading of employee information
    e1.get_PersonalInfo();
    e1.get_QualificationInfo();
    // showing of employee information
    e1.show_PersonalInfo();
    e1.show_QualificationInfo();

}
}

```



Regd No:

--	--	--	--	--	--	--	--	--	--

Output:

```

D:\Java_Programs>java Employee
Enter your empid, name, desg, salary, organization name, address
1111
Ramesh S
Assistant Professor
35000
Aditya Engineering College
Kakinada
Enter your UG course, marks and PG course and Marks
B.Tech CSE
75
M.Tech CSE
82

```

```

=====
PERSONAL INFORMATION
=====

```

```

EMPID: 1111
EMP NAME: Ramesh S
DESIGNATION: Assistant Professor
SALARY: 35000.0
ORGANIZATION NAME: Aditya Engineering College
ADDRESS: Kakinada

```

```

=====
QUALIFICATION INFORMATION
=====

```

```

UG COURSE: B.Tech CSE
UG PERCENTAGE: 75.0
PG COURSE: M.Tech CSE
PG PERCENTAGE: 82.0

```

Regd No:

--	--	--	--	--	--	--	--	--	--

4.2) Write a Java program to implement constructor and constructor overloading.**Source Code:**

```

class Box_Demo
{
    int l,b,area;

    public Box_Demo()                // Default Constructor
    {
        System.out.println("Default");
        l=b=1;
    }
    public Box_Demo(int m)           // Parameterized
    {
        System.out.println("SQUARE");
        l=m;
        b=m;
    }
    public Box_Demo(int m,int n)     // Parameterized
    {
        System.out.println("RECTANGLE");
        l=m;
        b=n;
    }
    public void Cal_Area()
    {
        area=l*b;
        System.out.println("Area is: "+area);
    }
    public static void main(String args[])
    {
        Box_Demo b1=new Box_Demo(5);
        b1.Cal_Area();

        Box_Demo b2=new Box_Demo(3,4);
        b2.Cal_Area();

        Box_Demo b3=new Box_Demo();
        b3.Cal_Area();
    }
}

```

Output:

```

D:\Java_Programs>java Box_Demo
SQUARE
Area is: 25
RECTANGLE
Area is: 12
Default
Area is: 1

```

Regd No:

--	--	--	--	--	--	--	--	--	--

4.3) Write a Java program to implement method overloading.**Source Code:**

```

class Method_Overloading
{
    public void methodOne()
    {
        System.out.println("no argument");
    }
    public void methodOne(int x,int y)
    {
        System.out.println(x+y);
    }
    public void methodOne(int d)
    {
        System.out.println(d);
    }
    public void methodOne(double d)
    {
        System.out.println(d);
    }
    public static void main(String args[])
    {
        Method_Overloading mo=new Method_Overloading();

        mo.methodOne();
        mo.methodOne(10);
        mo.methodOne(10,20);
        mo.methodOne(3.14);
    }
}

```

Output:

D:\Java_Programs>javac Method_Overloading.java

D:\Java_Programs>java Method_Overloading

no argument

10

30

3.14

Regd No:

--	--	--	--	--	--	--	--	--	--

Experiment No: 5**Date:****Aim: 5) Working with Strings****5.1) Write a Java program to sort given set of strings.****Source Code:**

```

import java.util.*;
public class String_Sort
{
    public static void main(String[] args)
    {
        int count;
        String temp;
        Scanner scan = new Scanner(System.in);

        //User will be asked to enter the count of strings
        System.out.println("Enter number of strings you would like to enter:");
        count = scan.nextInt();

        String str[] = new String[count];

        //User is entering the strings and they are stored in an array
        System.out.println("Enter the Strings one by one:");
        scan.nextLine();
        for(int i = 0; i < count; i++)
        {
            str[i] = scan.nextLine();
        }

        //Sorting the strings
        for (int i = 0; i < count; i++)
        {
            for (int j = i + 1; j < count; j++) {
                if (str[i].compareTo(str[j])>0)
                {
                    temp = str[i];
                    str[i] = str[j];
                    str[j] = temp;
                }
            }
        }

        //Displaying the strings after sorting them based on alphabetical order
        System.out.print("Strings in Sorted Order:");
        for (int i = 0; i <= count - 1; i++)
        {
            System.out.print(str[i] + ", ");
        }
    }
}

```

Regd No:

--	--	--	--	--	--	--	--	--	--

Output:

D:\>java String_Sort

Enter number of strings you would like to enter:

5

Enter the Strings one by one:

Rama

Sita

Laxman

Hanuma

Bharata

Strings in Sorted Order: Bharata, Hanuma, Laxman, Rama, Sita



Regd No:

--	--	--	--	--	--	--	--	--	--

5.2) Write a Java program for using String Buffer to remove or delete a character.

Source Code:

```
class StringBuffer_Demo
{
    public static void main(String args[])
    {
        StringBuffer s1=new StringBuffer();
        System.out.println(s1.capacity());    // 16 => C=(S+1)*2 , 34
        System.out.println(s1.length());      // 0
        StringBuffer s2=new StringBuffer("Welcome ");
        System.out.println(s2.capacity());    // 24

        System.out.println(s2.charAt(4));    // o

        s2.setCharAt(4,'a');
        System.out.println(s2); // Welcame

        s2.deleteCharAt(4);
        System.out.println(s2);    // Welcme

        s2.append(" Srinu");
        System.out.println(s2);    // Welcme Srinu

        s2.insert(4,"a");
        System.out.println(s2);    // Welcame Srinu

        s2.delete(8,13);
        System.out.println(s2);    // Welcame

        s2.append(true);
        System.out.println(s2);    // Welcame true

        s2.reverse();
        System.out.println(s2);    // eurt emacleW
    }
}
```

Output:

D:\>java StringBuffer_Demo

```
16
0
24
o
Welcame
Welcme
Welcme Srinu
Welcame Srinu
Welcame u
Welcame utrue
eurtu emacleW
```

Regd No:

--	--	--	--	--	--	--	--	--	--

5.3) Write a Java program to find the number of tokens in a given string without using count-Tokens() method but by using other methods of String Tokenizer class.

Source Code:

```
import java.util.StringTokenizer;
public class String_TokenizerDemo
{
    public static void main(String args[])
    {
        StringTokenizer st = new StringTokenizer("my name is khan and your name is
            salman"," ");
        //System.out.println(st.countTokens());
        int count=0;
        while (st.hasMoreTokens())
        {
            System.out.println(st.nextToken()); // my nam
            count++;
        }

        System.out.println("No of Tokens: "+count);
    }
}
```

Output:

```
D:\>java String_TokenizerDemo
my
name
is
khan
and
your
name
is
salman
No of Tokens: 9
```



Regd No:

--	--	--	--	--	--	--	--	--	--

Experiment No: 6**Date:****Aim: 6) Working with Inheritance, Interface & Abstract Class****6.1) Write a Java program to implement Single Inheritance.****Source Code:**

```

class A
{
    int x=10;
    public void showX()
    {
        System.out.println("X = "+x);
    }
}
class B extends A
{
    int y=20;
    public void showY()
    {
        System.out.println("Y = "+y);
    }
}
class SingleLevel
{
    public static void main(String args[])
    {
        A a=new A();
        a.showX();

        System.out.println("=====");
        B b=new B();
        b.showX();
        b.showY();
    }
}

```

Output:

D:\Practice\Inheritance>java SingleLevel

X = 10

=====

X = 10

Y = 20

Regd No:

--	--	--	--	--	--	--	--	--	--

6.2) Write a Java program to find the available balance in a customer account details should be taken in another class. (Note: Make use of Multi-Level Inheritance.)

Source Code:

```
import java.util.*;
class Customer
{
    String cust_id, cust_name, address;
    float balance;
    Scanner sc=new Scanner(System.in);

    public void get_CustomerInfo()
    {
        System.out.println("Enter Customer ID, Name, Balance and address");
        cust_id=sc.nextLine();
        cust_name=sc.nextLine();
        balance=sc.nextFloat();
        sc.nextLine();
        address=sc.nextLine();
    }
    public void show_CustomerInfo()
    {
        System.out.println("Customer Details are:");
        System.out.println("Id: "+cust_id);
        System.out.println("Name: "+cust_name);
        System.out.println("Balance: "+balance);
        System.out.println("Address: "+address);
    }
}
class Transaction extends Customer
{
    public void deposit(float amt)
    {
        System.out.println("Amount Deposited: "+amt);
        balance=balance+amt;
    }
    public void withdraw(float amt)
    {
        System.out.println("Amount withdrawn: "+amt);
        balance=balance-amt;
    }
    public void show_Bal()
    {
        System.out.println("Available Balance: "+balance);
    }
}
```

Regd No:

--	--	--	--	--	--	--	--	--	--

```

    }
}
class Bank extends Transaction
{
    static String bankname="Canara Bank",ifsc="CBN0003268";

    public void show_BankInfo()
    {
        System.out.println("Bank Name: "+Bank.bankname);
        System.out.println("IFSC Code: "+Bank.ifsc);
    }

    public static void main(String args[])
    {
        Bank c1=new Bank();

        c1.get_CustomerInfo();
        c1.show_CustomerInfo();
        c1.show_BankInfo();

        c1.deposit(10000);
        c1.show_Bal();

        c1.withdraw(5000);
        c1.show_Bal();
    }
}

```



Regd No:

--	--	--	--	--	--	--	--	--	--

Output:

D:\Inheritance>java Bank

Enter Customer ID, Name, Balance and address

32682210001421

M.Srinu

200000

Kakinada

Customer Details are:

Id: 32682210001421

Name: M.Srinu

Balance: 200000.0

Address: Kakinada

Bank Name: Canara Bank

IFSC Code: CBN0003268

Amount Deposited: 10000.0

Available Balance: 210000.0

Amount withdrawn: 5000.0

Available Balance: 205000.0



Regd No:

--	--	--	--	--	--	--	--	--	--

6.3) Write a Java program to find the areas of different shapes using abstract classes.**Source Code:**

```

import java.util.*;
abstract class Shape
{
    Scanner sc=new Scanner(System.in);
    float s1,s2,a;
    final float pi=3.14f;
    public abstract void get_Input();
    public abstract void Cal_Area();
    public void show_Area()
    {
        System.out.println("Area is:"+a);
    }
}
class Rect extends Shape
{
    public void get_Input()
    {
        System.out.println("Enter L and B values");
        s1=sc.nextFloat();
        s2=sc.nextFloat();
    }
    public void Cal_Area()
    {
        a=s1*s2;
    }
}
class Circle extends Shape
{
    public void get_Input()
    {
        System.out.println("Enter radius of the Circle");
        s1=sc.nextFloat();
    }
    public void Cal_Area()
    {
        a=pi*s1*s1;
    }
}

```

Regd No:

--	--	--	--	--	--	--	--	--	--

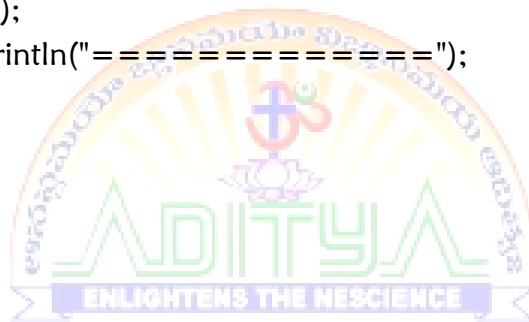
```

class Mainclass
{
    public static void main(String args[])throws Exception
    {
        Shape s;

        s=new Rect();
        System.out.println("Rectangle:");
        s.get_Input();
        s.Cal_Area();
        s.show_Area();
        System.out.println("=====");

        s=new Circle();
        System.out.println("Circle:");
        s.get_Input();
        s.Cal_Area();
        s.show_Area();
        System.out.println("=====");
    }
}

```



Regd No:

--	--	--	--	--	--	--	--	--	--

Output:

D:\Inheritance>java Mainclass

Rectangle:

Enter L and B values

14

16

Area is:224.0

=====

Circle:

Enter radius of the Circle

5.6

Area is:98.4704

=====



Regd No:

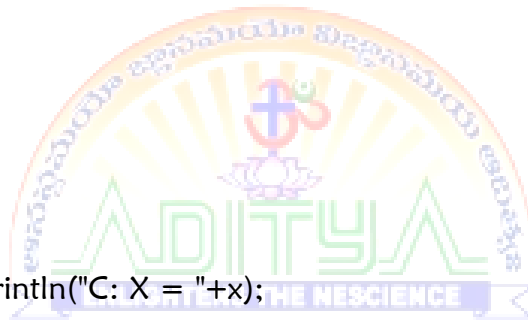
--	--	--	--	--	--	--	--	--	--

Experiment No: 8**Date:****7.1) Write a Java program for “super” keyword.****Source Code:**

```

class A
{
    int x=10;
    public void show()
    {
        System.out.println("A: X = "+x);
    }
}
class B extends A
{
    int x=20;
    public void show()
    {
        super.show();
        System.out.println("B: X = "+x);
    }
}
class C extends B
{
    int x=30;
    public void show()
    {
        super.show();
        System.out.println("C: X = "+x);
    }
}
class MainDemo
{
    public static void main(String args[])
    {
        C c=new C();
        c.show();
    }
}

```

**Output:**

D:\Practice>java MainDemo

A: X = 10

B: X = 20

C: X = 30

Regd No:

--	--	--	--	--	--	--	--	--	--

7.2) Take the details of internal exam marks in one Interface. Take the details of external exam marks in another interface. Write a Java program to find the total marks obtained in each subject by a student. (Note: Make use of Multiple Inheritance using interfaces.).

Source Code:

```
import java.util.*;
interface Internal
{
    void get_InternalMarks();
}
interface External
{
    void get_ExternalMarks();
}
interface Marks extends Internal, External
{
    void show_Marks();
}
class Result implements Marks
{
    // s1_i -> sub1 internal, s1_e -> sub1 external
    float s1_i,s1_e,s2_i,s2_e,s3_i,s3_e;
    Scanner sc=new Scanner(System.in);
    public void get_InternalMarks()
    {
        System.out.println("Enter 3 subjects internal marks (0 - 40)");
        s1_i=sc.nextFloat();
        s2_i=sc.nextFloat();
        s3_i=sc.nextFloat();
    }
    public void get_ExternalMarks()
    {
        System.out.println("Enter 3 subjects External marks (0 - 60)");
        s1_e=sc.nextFloat();
        s2_e=sc.nextFloat();
        s3_e=sc.nextFloat();
    }
    public void show_Marks()
    {
        System.out.println("Subject \t Internal\tExternal\tTotal_Marks: ");
        System.out.println(" Sub1 \t\t "+s1_i+"\t\t"- Regd No: "+(s1_i+s1_e));
        System.out.println(" Sub2 \t\t "+s2_i+"\t\t"+s2_e+"\t\t"+(s2_i+s2_e));
        System.out.println(" Sub3 \t\t "+s3_i+"\t\t"+s3_e+"\t\t"+(s3_i+s3_e));
    }
}
```

Regd No:

--	--	--	--	--	--	--	--	--	--

```

}

class Mainclass
{
    public static void main(String args[])
    {
        Marks m=new Result();
        m.get_InternalMarks();
        m.get_ExternalMarks();
        m.show_Marks();
    }
}

```

Output:

D:\Inheritance>java Mainclass
Enter 3 subjects internal marks (0 - 40)
37 29 38
Enter 3 subjects External marks (0 - 60)
57 49 58

Subject	Internal	External	Total_Marks:
Sub1	37.0	57.0	94.0
Sub2	29.0	49.0	78.0
Sub3	38.0	58.0	96.0



Regd No:

--	--	--	--	--	--	--	--	--	--

Experiment No: 8**Date:****8.1) Write a JAVA program that implements Runtime polymorphism****Dynamic Method Dispatching:**

Single parent class object is overloaded with different child class instance at different times, and it will invoke the related methods according to child class instance, which is held in that object at runtime. This is called “Dynamic Method Dispatching”.

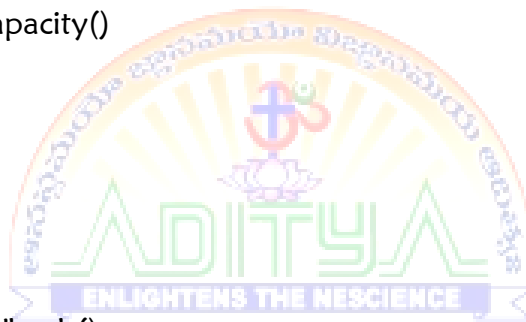
Source Code:

```
abstract class Vehicle
{
    public abstract int get_No_Wheels();
    public abstract int seating_Capacity();
}
```

```
class Bike extends Vehicle
{
    public int get_No_Wheels()
    {
        return 2;
    }
    public int seating_Capacity()
    {
        return 2;
    }
}
```

```
class Auto extends Vehicle
{
    public int get_No_Wheels()
    {
        return 3;
    }
    public int seating_Capacity()
    {
        return 4;
    }
}
```

```
class Car extends Vehicle
{
    public int get_No_Wheels()
    {
        return 4;
    }
    public int seating_Capacity()
    {
        return 5;
    }
}
```

**Regd No:**

--	--	--	--	--	--	--	--	--	--

```

class Mainclass
{
    public static void main(String args[])
    {
        Vehicle v;
        int w,c;
        v=new Bike();
        System.out.println("=====");
        System.out.println("Bike:");
        w=v.get_No_Wheels();
        c=v.seating_Capacity();
        System.out.println("No of Wheels: "+w);
        System.out.println("Seating Capacity: "+c);
        System.out.println("=====");

        v=new Auto();
        System.out.println("=====");
        System.out.println("Auto:");
        w=v.get_No_Wheels();
        c=v.seating_Capacity();
        System.out.println("No of Wheels: "+w);
        System.out.println("Seating Capacity: "+c);
        System.out.println("=====");

        v=new Car();
        System.out.println("=====");
        System.out.println("Car:");
        w=v.get_No_Wheels();
        c=v.seating_Capacity();
        System.out.println("No of Wheels: "+w);
        System.out.println("Seating Capacity: "+c);
        System.out.println("=====");
    }
}

```

Output:

D:\Java_Programs>java Mainclass

=====

Bike:

No of Wheels: 2

Seating Capacity: 2

=====

=====

Auto:

No of Wheels: 3

Seating Capacity: 4

=====

=====

Car:

No of Wheels: 4

Regd No:

--	--	--	--	--	--	--	--	--	--

Seating Capacity: 5

Experiment No: 9**Date:****Aim: 9) Working with Packages****9.1) Write a Java program that import and use user defined package.****User defined packages**

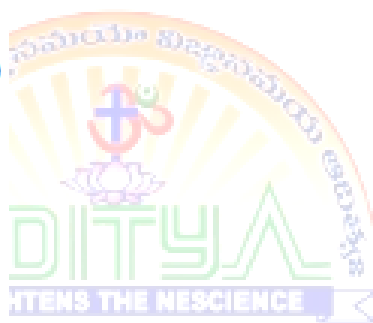
User-defined packages are those which are developed by users in order to group related classes, interfaces and sub packages. With the help of an example program, let's see how to create packages, compile Java programs inside the packages and execute them.

Steps involved in user defined package creation:

1. Creation of user defined package file
2. Compilation of user defined package file
3. Setting of class path
4. Importing of user defined package in another application.

Source Code:**Step – 1:**

```
package myPack;
public class Compare
{
    public int getMax(int n, int m)
    {
        if(n>m)
            return n;
        else
            return m;
    }
    public int getMin(int n,int m)
    {
        if(n<m)
            return n;
        else
            return m;
    }
    public void getEqual(int n,int m)
    {
        if(n==m)
            System.out.println("Equal");
        else
            System.out.println("Not Equal");
    }
}
```

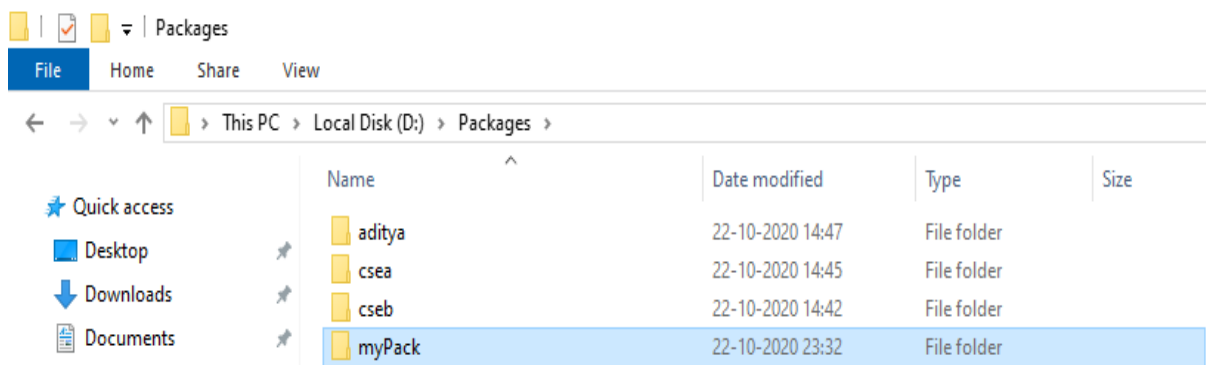
**Step – 2:**

```
D:\Java_Programs>javac -d D:\Packages Compare.java
```

```
D:\Java_Programs>
```

Regd No:

--	--	--	--	--	--	--	--	--	--

**Step – 3 & 4:**

```

1 import java.util.Scanner;    // built in package
2 import myPack.Compare;      // user defined package
3 class NumberDemo
4 {
5     public static void main(String args[])
6     {
7         int n1,n2;
8         Scanner sc=new Scanner(System.in);
9         Compare c=new Compare();
10        System.out.println("Enter any two numbers");
11        n1=sc.nextInt();
12        n2=sc.nextInt();
13
14        int max=c.getMax(n1,n2);
15        int min=c.getMin(n1,n2);
16        c.getEqual(n1,n2);
17
18        System.out.println("Maximum: "+max);
19        System.out.println("Minimum: "+min);
20    }
21 }

```

Output:

```
D:\Java_Programs>set classpath=.;D:\Packages;%classpath%;
```

```
D:\Java_Programs>javac NumberDemo.java
```

```
D:\Java_Programs>java NumberDemo
```

```
Enter two numbers
```

```
25
```

```
36
```

```
Not Equal
```

```
Maximum value is:36
```

```
Minimum value is:25
```

Regd No:

--	--	--	--	--	--	--	--	--	--

9.2) Write a Java program to illustrate the use of protected members in a package.

Protected: The protected access modifier is accessible within package and outside the package but through inheritance only.

The protected access modifier can be applied on the data member, method and constructor. It can't be applied on the class.

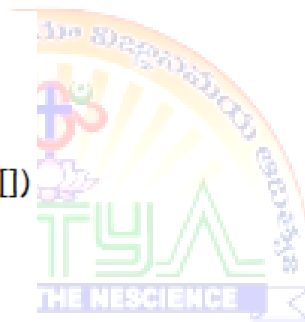
Source Code:**CASE – 1:**

```

1 package pack1;
2 public class A
3 {
4     protected void m1()
5     {
6         System.out.println("protected method in A");
7     }
8 }

1 package pack2;
2 import pack1.A;
3 public class B extends A
4 {
5     public static void main(String args[])
6     {
7         B b=new B();
8         b.m1();
9     }
10 }

```

**Output:**

```
D:\package>javac -d D:\Prog A.java
```

```
D:\package>javac -d D:\Prog B.java
```

```
D:\package>java pack2.B
protected method in A
```

Regd No:

--	--	--	--	--	--	--	--	--	--

CASE – 2:

```

1 package pack1;
2 public class A
3 {
4     protected void m1()
5     {
6         System.out.println("protected method in A");
7     }
8 }

1 package pack2;
2 import pack1.A;
3 public class B
4 {
5     public static void main(String args[])
6     {
7         A a=new A();
8         a.m1();
9     }
10 }

```

Output:

D:\package>javac -d D:\Prog A.java

D:\package>javac -d D:\Prog B.java

B.java:8: error: m1() has protected access in A
 a.m1();
 ^

1 error

Regd No:

--	--	--	--	--	--	--	--	--	--

Experiment No: 10**Date:****Aim: 10) Working with Exception Handling****10.1) Write a Java program to illustrate exception handling mechanism using multiple catch clauses.****Source Code:**

```

class ExceptionDemo
{
    public static void main(String[] args)
    {
        int m, n, o=0;
        try
        {
            m = Integer.parseInt(args[0]);
            n = Integer.parseInt(args[1]);
            o = m/n;
        }
        catch(ArrayIndexOutOfBoundsException ae)
        {
            System.out.println(ae.getMessage());
        }
        catch(NumberFormatException ne)
        {
            System.out.println(ne.getMessage());
        }
        catch(ArithmeticException are)
        {
            are.printStackTrace();
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
        finally
        {
            System.out.println("Cleanup code");
            System.out.println(o);
        }
    }
}

```

Regd No:

--	--	--	--	--	--	--	--	--	--

Output:

D:\>java ExceptionDemo 10 2

Cleanup code

5

D:\>java ExceptionDemo 10

Index 1 out of bounds for length 1

Cleanup code

0

D:\>java Exception1 10 0

java.lang.ArithmeticException: / by zero at Exception1.main(Exception1.java:9)

Cleanup code

0

D:\>java Exception1 10 a

For input string: "a"

Cleanup code

0



Regd No:

--	--	--	--	--	--	--	--	--	--

10.2) Write a Java program to make use of Built-in and user-defined Exceptions in handling a run.

Problem Statement: Write a Program to take care of NumberFormatException if user enters values other than integer for calculating average marks of a student. The name of the student and marks in 3 subjects are taken from user while executing the program. In the same program write your own Exception classes to take care of **Negative values** and values out of range(i.e other than in the range of 0 -100).

Source Code:

```
class NegativeValException extends Exception
{
    public NegativeValException(String msg)
    {
        super(msg);
    }
}
class Excep3
{
    public static void main(String args[])
    {
        String name=null;
        int m1=0,m2=0,m3=0;
        try
        {
            name=args[0];
            m1=Integer.parseInt(args[1]);
            m2=Integer.parseInt(args[2]);
            m3=Integer.parseInt(args[3]);
            if(m1<0 || m2<0 || m3<0)
                throw new NegativeValException("Marks should be greater than 0");
        }
        catch(ArrayIndexOutOfBoundsException aoe)
        {
            System.out.println("Minimum of 4 arguments you need to pass");
        }
        catch(NumberFormatException ne)
        {
            System.out.println("Marks should be Integers only");
        }
        catch(NegativeValException nve)
        {
            System.out.println("Marks should be greater than zero");
            System.exit(0);
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
        System.out.println("Name = "+name);
        System.out.println("Average Marks="+((m1+m2+m3)/3));
    }
}
```

Output:

D:\Practice>java Excep3 Ramesh 45 78 65

Name = Ramesh

Average Marks=62

D:\Practice>java Excep3 Ramesh 45 78

Minimum of 4 arguments you need to pass

Name = Ramesh

Average Marks=41

D:\Practice>java Excep3 Ramesh 45 78 -52

Marks should be greater than zero



Regd No:

--	--	--	--	--	--	--	--	--	--

Experiment No: 11**Date:****Aim: 11) Working with Multithreading****11.1) Write a Java program to demonstrate the use of demon thread.****Source Code:**

```

class DaemonThreadExample extends Thread
{
    public void run()
    {
        // Checking whether the thread is Daemon or not
        if(Thread.currentThread().isDaemon())
        {
            System.out.println("Daemon thread executing");
        }
        else
        {
            System.out.println("user(normal) thread executing");
        }
    }

    public static void main(String[] args)
    {
        /* Creating two threads: by default they are user threads
        (non-daemon threads)
        */
        DaemonThreadExample t1=new DaemonThreadExample();
        DaemonThreadExample t2=new DaemonThreadExample();

        //Making user thread t1 to Daemon
        t1.setDaemon(true);

        //starting both the threads
        t1.start();
        t2.start();
    }
}

```

Output:

```

D:\java_prog\MultiThreading\Threading_Prog1>java DaemonThreadExample
Daemon thread executing
user(normal) thread executing

```

Regd No:

--	--	--	--	--	--	--	--	--	--

11.2) Write a Java program that creates threads by extending Thread class. First thread display “Good Morning” every 1 sec, the second thread displays “Hello” every 2 seconds and the third display “Welcome” every 3 seconds, (Repeat the same by implementing Runnable).

Source Code:

```
class MyThread1 extends Thread
```

```
{
    public void run()
    { try{
        while(true)
        {
            System.out.println(Thread.currentThread().getName()+" : Good Morning");
            Thread.sleep(1000);
        }
    }
    catch(InterruptedException ie)
    {
    }
}
}
```

```
class MyThread2 extends Thread
```

```
{
    public void run()
    {
        try{
            while(true)
            {
                System.out.println(Thread.currentThread().getName()+" : Hello");
                Thread.sleep(2000);
            }
        }
        catch(InterruptedException ie)
        {
        }
    }
}
}
```

```
class MyThread3 extends Thread
```

```
{
    public void run()
    {
        try{
            while(true)
            {
                System.out.println(Thread.currentThread().getName()+" : Welcome");
                Thread.sleep(3000);
            }
        }
    }
}
```

Regd No:

--	--	--	--	--	--	--	--	--	--

```


        }

    }
    catch(InterruptedException ie)
    {
    }
}
}
class MainDemo
{
    public static void main(String args[])
    {
        MyThread1 t1=new MyThread1();
        MyThread2 t2=new MyThread2();
        MyThread3 t3=new MyThread3();
        t1.setName("A");
        t2.setName("B");
        t3.setName("C");

        t1.start();
        t2.start();
        t3.start();

        System.out.println("MAIN CLOSED");
    }
}

```

Output:


```

D:\java_prog\MultiThreading>java MainDemo
MAIN CLOSED
C: Welcome
B: Hello
A: Good Morning
A: Good Morning
B: Hello
A: Good Morning
C: Welcome
A: Good Morning
B: Hello
A: Good Morning
A: Good Morning
C: Welcome
B: Hello
A: Good Morning
A: Good Morning

```

Regd No:

--	--	--	--	--	--	--	--	--	--

11.3) Write a Java program to solve Producer-Consumer problem using synchronization.**Source Code:**

```

import java.util.*;
//critical section: buffer object is passed into both producer and consumer threads
class Buffer
{
    String data;
    boolean avail=false;
    public synchronized void put(String data)    //used by producer
    {
        while (avail==true)
        {
            try
            {
                wait ();
            }
            catch (InterruptedException ie)
            {System.out.println(ie);}
        }
        this.data=data;
        System.out.println("Produced:"+data);
        avail = true;
        notify();
    }
    public synchronized String get()    // used by consumer
    {
        while(avail==false)
        {
            try
            {
                wait ();    // it throws InterruptedException
            }
            catch (InterruptedException ie)
            {System.out.println(ie);}
        }
        avail = false;
        notify ();
        return data;
    }
}
class Producer extends Thread
{
    String data;
    Scanner sc=new Scanner(System.in);
    Buffer buf;
    public Producer(Buffer buf)
    {
        super("Producer");
        this.buf=buf;
    }
}

```

Regd No:

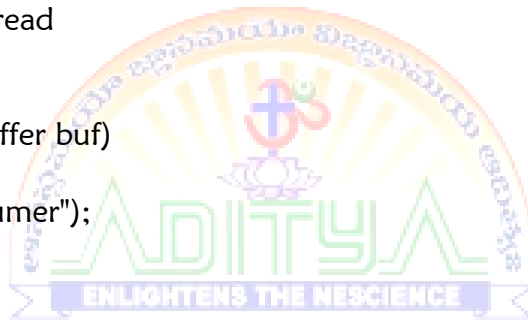
--	--	--	--	--	--	--	--	--	--

```

    }
    public void run ()
    {
        try
        {
            while (true)
            {
                System.out.println("Enter data");
                data=sc.nextLine();
                buf.put(data);
                Thread.sleep(500);           //InterruptedException
            }
        }
        catch (InterruptedException e)
        {
            System.out.println(e);
        }
    }
}

class Consumer extends Thread
{
    Buffer buf;
    public Consumer(Buffer buf)
    {
        super ("Consumer");
        this.buf=buf;
    }
    public void run ()
    {
        try
        {
            while (true)
            {
                System.out.println("Consumed:"+ buf.get());
                Thread.sleep (500);
            }
        }
        catch(InterruptedException e)
        {
            System.out.println(e);
        }
    }
}

```



Regd No:

--	--	--	--	--	--	--	--	--	--

```

class MainDemo
{
    public static void main (String args [])
    {
        Buffer buf = new Buffer ();
        Producer p = new Producer (buf);
        Consumer c = new Consumer (buf);

        p.start();
        c.start();
    }
}

```

Output:

D:\java_prog\MultiThreading>java MainDemo

Enter data

CSE

Produced:CSE

Consumed:CSE

Enter data

ECE

Produced:ECE

Consumed:ECE

Enter data

EEE

Produced:EEE

Consumed:EEE

Enter data

MECH

Produced:MECH

Consumed:MECH

Enter data



Regd No:

--	--	--	--	--	--	--	--	--	--

Experiment No: 12**Date:**

Write a JDBC program to perform the following operations by connecting to MYSQL database.

I) Inserting Data into Table

II) Updating Data in the Table.

III) Deleting Data From the Table based on a column value.

Table Creation:

```
import java.sql.*;
class Table_Creation
{
    public static void main(String args[])
    {
        try{
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/db1","root","Thub@123");
            if(con!=null)
                System.out.println("Database Connected");
            else
                System.out.println("Not Connected");

            Statement st=con.createStatement();
            int count=st.executeUpdate("create table movie(mno int, mname varchar(20), Hero
            varchar(20), Heroine varchar(30))");
            if(count==1)
                System.out.println("table created");
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
    }
}
```

Output:

```
D:\java_prog\JDBC\JDBC Programs>java Table_Creation
Database Connected
Table created
```

Insertion of data into a table:

```

import java.sql.*;
import java.io.*;
class Dynamic_Entry
{
    public static void main(String args[])
    {
        try{
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection
            con=DriverManager.getConnection("jdbc:mysql://localhost:3306/db1","root","Thub@123");
            if(con!=null)
                System.out.println("Database Connected");
            else
                System.out.println("Not Connected");

            Statement st=con.createStatement();
            DataInputStream din=new DataInputStream(System.in);
            while(true)
            {
                System.out.println("Enter Movie no:");
                int no=Integer.parseInt(din.readLine());

                System.out.println("Enter MovieName name:");
                String name=din.readLine();

                System.out.println("Enter Hero name:");
                String hero=din.readLine();

                System.out.println("Enter Heroine name:");
                String heroine=din.readLine();

                int count=st.executeUpdate("insert into movie values("+no+", '"+name+"', '"+hero+"', '"+heroine+"'");
                if(count==1)
                    System.out.println("Record Inserted Successfully");

                System.out.println("Do you want to insert another record [yes/no]");
                String choice=din.readLine();
                if(choice.equalsIgnoreCase("no"))
                    break;
            }
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
    }
}

```

Output:

```
D:\java_prog\JDBC\JDBC Programs>java Dynamic_Entry
Database Connected
Enter Movie no:
101
Enter MovieName name:
RRR
Enter Hero name:
RamCharan
Enter Heroine name:
Alia Bhatt
Record Inserted Successfully
Do you want to insert another record [yes/no]
yes
Enter Movie no:
102
Enter MovieName name:
F3
Enter Hero name:
Venkatesh
Enter Heroine name:
Tamanna
Record Inserted Successfully
Do you want to insert another record [yes/no]
yes
Enter Movie no:
103
Enter MovieName name:
Acharya
Enter Hero name:
Chiranjeevi
Enter Heroine name:
Pooja
Record Inserted Successfully
Do you want to insert another record [yes/no]
yes
Enter Movie no:
104
Enter MovieName name:
Bahubali
Enter Hero name:
Prabhas
Enter Heroine name:
Anushka
Record Inserted Successfully
Do you want to insert another record [yes/no]
no
```



Selection of data from a table:

```

import java.sql.*;
class Select_Demo
{
    public static void main(String args[])
    {
        try{
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/db1","root","Thub@123");
            if(con!=null)
                System.out.println("Database Connected");
            else
                System.out.println("Not Connected");

            Statement st=con.createStatement();
            ResultSet rs=st.executeQuery("select * from movie");
            while(rs.next())
            {
                int no=rs.getInt(1);
                //System.out.print(no);
                String name=rs.getString(2);
                //System.out.println("---"+name);
                String hero=rs.getString(3);
                String heroine=rs.getString(4);

                System.out.println(no+"-----"+name+"-----"+hero+"-----"+heroine);
            }
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
    }
}

```

Output:

D:\java_prog\JDBC\JDBC Programs>java Select_Demo

Database Connected

101 -----RRR-----RamCharan-----Alia Bhatt

102 -----F3-----Venkatesh-----Tamanna

103 -----Acharya-----Chiranjeevi-----Pooja

104 -----Bahubali-----Prabhas-----Anushka

Updating a record in a table:

```

import java.sql.*;
class Update_Demo
{
    public static void main(String args[])
    {
        try{
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/db1","root","Thub@123");
            if(con!=null)
                System.out.println("Database Connected");
            else
                System.out.println("Not Connected");

            Statement st=con.createStatement();
            int count=st.executeUpdate("update movie set Hero='Ram Charan' where mno=103");
            if(count==1)
                System.out.println("Record Updated Successfully");
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
    }
}

```

Output:

D:\java_prog\JDBC\JDBC Programs>java Update_Demo

Database Connected

Record Updated Successfully

Deletion of a record from a table:

```

import java.sql.*;
class Delete_Demo
{
    public static void main(String args[])
    {
        try{
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/db1","root","Thub@123");
            if(con!=null)
                System.out.println("Database Connected");
            else
                System.out.println("Not Connected");

            Statement st=con.createStatement();
            int count=st.executeUpdate("delete from movie where mno=104");
            if(count>0)
                System.out.println("Record Deleted Successfully");
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
    }
}

```

Output:

D:\java_prog\JDBC\JDBC Programs>java Delete_Demo
Database Connected
Record Deleted Successfully



Augmented Experiments**Date:**

(Any 2 of the given experiments can be performed)

13) Write a Java program that simulates a traffic light. The program lets the user select one of three lights: red, yellow, or green with radio buttons. On selecting a button, an appropriate message with "Stop" or "Ready" or "Go" should appear above the buttons in selected color.

Source Code:

```
import java.awt.*;
import java.awt.event.*;
class TrafficLightsExample extends Frame implements ItemListener
{
    CheckboxGroup grp;
    Checkbox redLight, yellowLight, greenLight;
    Label msg;
    public TrafficLightsExample()
    {
        grp=new CheckboxGroup();
        redLight = new Checkbox("Red", grp, false);
        yellowLight = new Checkbox("Yellow", grp, false);
        greenLight = new Checkbox("Green", grp, false);
        msg = new Label("");

        setLayout(new FlowLayout());
        redLight.addItemListener(this);
        yellowLight.addItemListener(this);
        greenLight.addItemListener(this);

        add(redLight);
        add(yellowLight);
        add(greenLight);
        add(msg);
        msg.setFont(new Font("Serif", Font.BOLD, 20));
    }
    public void itemStateChanged(ItemEvent ie)
    {
        redLight.setForeground(Color.BLACK);
        yellowLight.setForeground(Color.BLACK);
        greenLight.setForeground(Color.BLACK);

        if(redLight.getState() == true)
        {
            redLight.setForeground(Color.RED);
            msg.setForeground(Color.RED);
            msg.setText("STOP");
        }
        else if(yellowLight.getState() == true)
        {
            yellowLight.setForeground(Color.YELLOW);
            msg.setForeground(Color.YELLOW);
            msg.setText("READY");
        }
    }
}
```

Regd No:

--	--	--	--	--	--	--	--	--	--

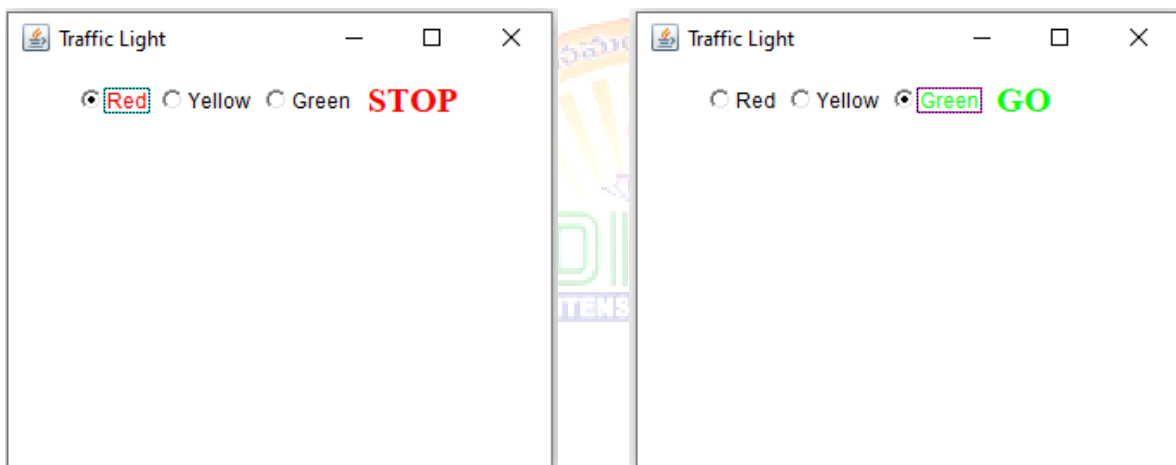
```

    }
    else
    {
        greenLight.setForeground(Color.GREEN);
        msg.setForeground(Color.GREEN);
        msg.setText("GO");
    }
}

public static void main(String args[])
{
    TrafficLightsExample tf=new TrafficLightsExample();
    tf.setSize(500,500);
    tf.setTitle("Traffic Light");
    tf.setVisible(true);
}
}

```

Output:



14) Develop a java program that receives an integer in one text field, and computes its factorial Value and returns it in another text field, when the button named “Compute” clicked.

Source Code:

```
import java.awt.*;
import java.awt.event.*;
class Factorial extends Frame implements ActionListener
{
    TextField input,output;
    Button compute;
    int fact=0;
    Label inp,opt;
    public Factorial()
    {
        setLayout(new FlowLayout());
        compute=new Button("Compute");
        inp=new Label("Enter any number :",Label.RIGHT);
        opt=new Label("Factorial of the given number is : ",Label.RIGHT);
        input=new TextField(5);
        output=new TextField(10);
        input.setBackground(Color.pink);
        output.setBackground(Color.pink);
        add(inp);
        add(input);
        add(opt);
        add(output);
        add(compute);
        output.setText("0");
        output.setEditable(false);
        input.addActionListener(this);
        output.addActionListener(this);
        compute.addActionListener(this);
    }
    public void actionPerformed(ActionEvent ae)
    {
        String str=ae.getActionCommand();
        if(str.equals("Compute"))
        {
            fact=1;
            int n=Integer.parseInt(input.getText());
            if(n<=12)
            {
                for(int i=n;i>=2;i--)
                fact=fact*i;
                output.setText(""+fact);
            }

            else
            fact=-1;
            output.setText(fact+"");
        }
    }
}
```



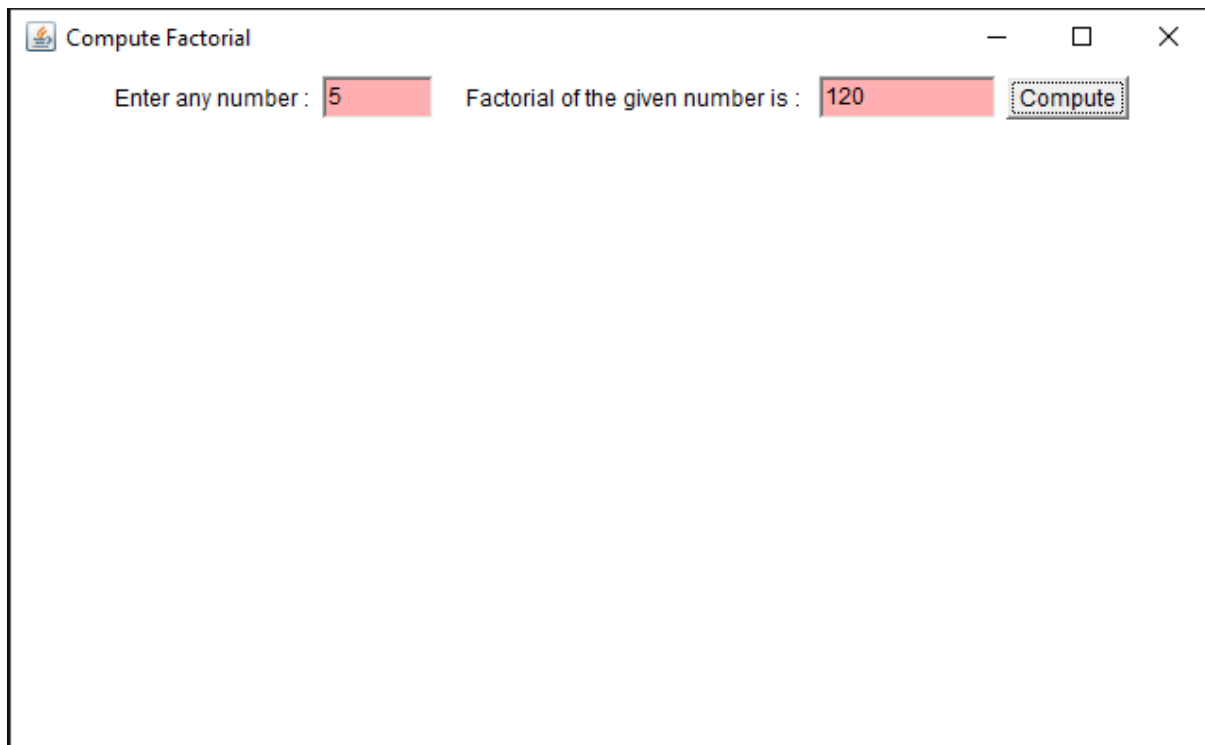
```
    }  
}  
public static void main(String args[])  
{  
    Factorial f=new Factorial();  
    f.setSize(400,400);  
    f.setTitle("Compute Factorial");  
    f.setVisible(true);  
}  
}
```



Regd No:

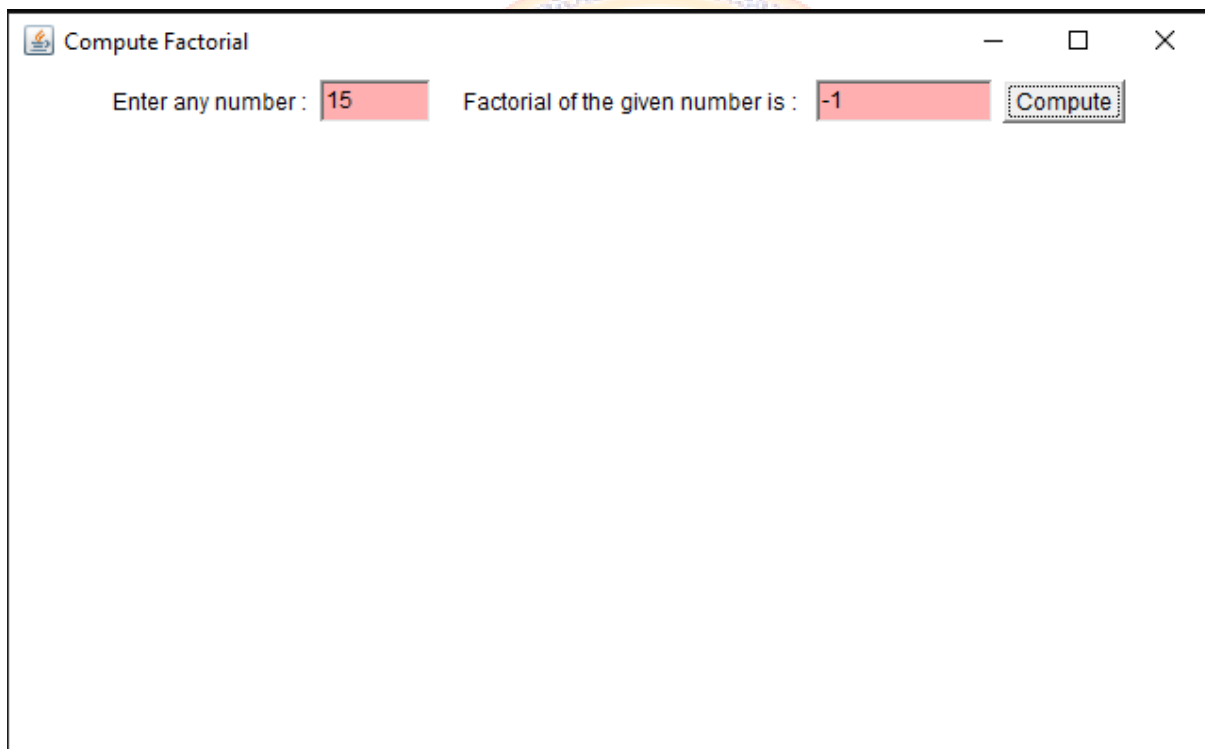
--	--	--	--	--	--	--	--	--	--

Output:



Compute Factorial

Enter any number : 5 Factorial of the given number is : 120



Compute Factorial

Enter any number : 15 Factorial of the given number is : -1

Regd No:

--	--	--	--	--	--	--	--	--	--