**Experiment-1: Basic Programs**

**1.1) Write a Java program to display default value of all primitive data type of JAVA**

**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

**1.2) Write a Java program to find the discriminant value D and find out the roots of**

**the quadratic equation of the form ax2+bx+c=0.**

**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);

        }

    }

}

**Output:**

Enter a, b and c values

1 4 4

root1 = root2 = -2.00;

**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.**

**Code:**

import java.util.\*;

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);

}

}

**Output:**

Enter 5 Bike Racers Speeds

60 70 80 90 100

Average Speed is:80.0

The Qualified Racers are:

90

100

**Experiment-2: Control Flow Statements**

**2.1) Write a Java program to select all the prime numbers within the range of 1 to 100.**

**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:**

Enter a number to find prime numbers upto it

60

2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59

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

**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:**

Enter the n value

100

sum of all even terms in the Fibonacci sequence up to the given range 100 is: 44

**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.

**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:**

Enter a number

371

371 is Armstrong Number

**Experiment-3: Arrays**

**3.1) Write a Java program to implement binary search.**

**Code:**

import java.util.\*;

class BinarySearchExample{

public static void binarySearch(int arr[], int first, int last, int key){

while( first <= last ){

int mid = (first + last)/2;

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;

}

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);

}

}

**Output:**

Enter the number of elements

5

Enter 5 elements

1 2 3 4 5

Enter the number to search

6

Element is not found!

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

**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);

System.out.println("Array After Bubble Sort");

for(int i=0; i < arr.length; i++)

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

}

}

**Output:**

5

50 40 30 20 10

Array Before Bubble Sort

50 40 30 20 10

Array After Bubble Sort

10 20 30 40 50

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

**Code:**

class Merge\_Sort

{

public static void merge(int a[], int beg, int mid, int end)

{

int i,j,k;

i=beg;

j=mid+1;

int x=0;

int temp[]=new int[end-beg+1];

while(i<=mid && j<=end)

{

if(a[i]<=a[j])

temp[x++]=a[i++];

else

temp[x++]=a[j++];

}

while(i<=mid)

temp[x++]=a[i++];

while(j<=end)

temp[x++]=a[j++];

k=0;

for(int s=beg;s<=end;s++)

a[s]=temp[k++];

}

public static 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);

}

}

public static 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("");

}

}

**Output:**

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

**Experiment-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.**

**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);

}

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[]){

Employee e1=new Employee();

e1.get\_PersonalInfo();

e1.get\_QualificationInfo();

e1.show\_PersonalInfo();

e1.show\_QualificationInfo();

}

}

**Output:**

Enter your empid, name, desg, salary, organization name, address

1

Harsha

Programmer

100000

T-Hub

Aditya Engineering College, Surampalem

Enter your UG course, marks and PG course and Marks

B.Tech

86

M.S

88

============================

PERSONAL INFORMATION

============================

EMPID: 1

EMP NAME: Harsha

DESGINATION: Programmer

SALARY: 100000.0

ORGANIZATION NAME: T-Hub

ADDRESS: Aditya Engineering College, Surampalem

=============================

QUALIFICATION INFORMATION

=============================

UG COURSE: B.Tech

UG PERCENTAGE: 86.0

PG COURSE: M.S

PG PERCENTAGE: 88.0

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

**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:**

SQUARE

Area is: 25

RECTANGLE

Area is: 12

Default

Area is: 1

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

**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:**

no argument

10

30

3.14

**Experiment-5: Strings**

**5.1) Write a Java program to sort given set of strings.**

**Code:**

import java.util.\*;

public class String\_Sort{

public static void main(String[] args) {

int count;

String temp;

Scanner scan = new Scanner(System.in);

System.out.println("Enter number of strings you would like to enter:");

count = scan.nextInt();

String str[] = new String[count];

System.out.println("Enter the Strings one by one:");

scan.nextLine();

for(int i = 0; i < count; i++)

str[i] = scan.nextLine();

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;

}

}

}

System.out.print("Strings in Sorted Order:");

for (int i = 0; i <= count - 1; i++)

System.out.print(str[i] + ", ");

}

}

**Output:**

Enter number of strings you would like to enter:

5

Enter the Strings one by one:

Java

Is

Nice

Programming

Language

Strings in Sorted Order:Is, Java, Language, Nice, Programming,

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

**Code:**

class StringBuffer\_Demo

{

public static void main(String args[])

{

StringBuffer s1=new StringBuffer();

System.out.println(s1.capacity());

System.out.println(s1.length());

StringBuffer s2=new StringBuffer("Welcome ");

System.out.println(s2.capacity());

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

s2.setCharAt(4,'a');

System.out.println(s2);

s2.deleteCharAt(4);

System.out.println(s2);

s2.append("Harsha");

System.out.println(s2);

s2.insert(4,"a");

System.out.println(s2);

s2.delete(8,13);

System.out.println(s2);

s2.append(true);

System.out.println(s2);

s2.reverse();

System.out.println(s2);

}

}

**Output:**

16

0

24

o

Welcame

Welcme

Welcme Harsha

Welcame Harsha

Welcame a

Welcame atrue

eurta emacleW

**Experiment-6: Inheritance**

**6.1) Write a Java program to implement Single Inheritance.**

**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:**

X = 10

=========================

X = 10

Y = 20

**6.2) Write a Java program to implement multi-level Inheritance.**

**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);

}

}

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();

}

}

**Output:**

Enter Customer ID, Name, Balance and address

CB1234

Harsha

12775

Surampalem

Customer Details are:

Id: UB1234

Name: Harsha

Balance: 12775.0

Address: Surampalem

Bank Name: Canara Bank

IFSC Code: CBN0003268

Amount Deposited: 10000.0

Available Balance: 22775.0

Amount withdrawn: 5000.0

Available Balance: 17775.0

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

**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;

}

}

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("=============");

}

}

**Output:**

Rectangle:

Enter L and B values

5 4

Area is:20.0

=============

Circle:

Enter radius of the Circle

5

Area is:78.5

=============

**Experiment-7: Inheritance-continued**

**7.1) Write a Java program for “super” keyword.**

**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:**

A: X = 10

B: X = 20

C: X = 30

**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.)**

**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{

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"+s1\_e+"\t\t"+(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));

}

}

class Mainclass{

public static void main(String args[]){

Marks m=new Result();

m.get\_InternalMarks();

m.get\_ExternalMarks();

m.show\_Marks();

}

}

**Output:**

Enter 3 subjects internal marks (0 - 40)

38 39 40

Enter 3 subjects External marks (0 - 60)

60 59 58

Subject Internal External Total\_Marks:

Sub1 38.0 60.0 98.0

Sub2 39.0 59.0 98.0

Sub3 40.0 58.0 98.0

**7.3) 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”.**

**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;

}

}

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:**

==============

Bike:

No of Wheels: 2

Seating Capacity: 2

==============

==============

Auto:

No of Wheels: 3

Seating Capacity: 4

==============

==============

Car:

No of Wheels: 4

Seating Capacity: 5

==============

**Experiment-8: Packages**

**8.1) 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.

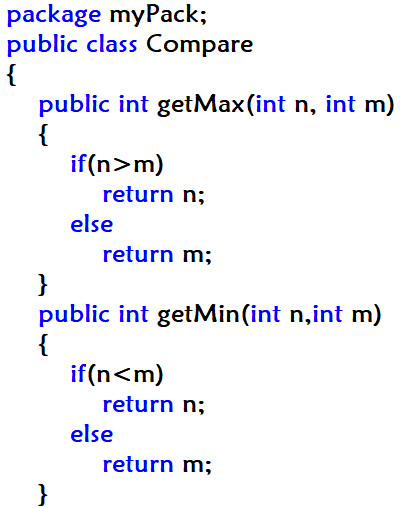
**Steps involved in user defined package creation:**

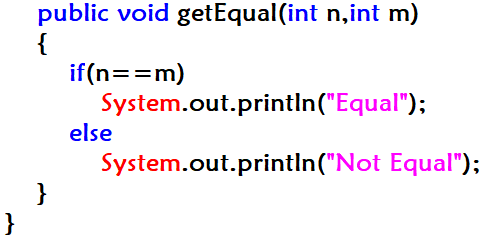
1. Creation of user defined package file
2. Compilation of user defined package file
3. Setting of class path

Importing of user defined package in another application.

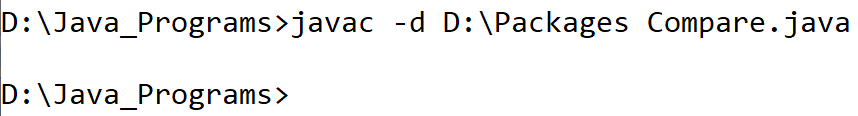
**Code:**

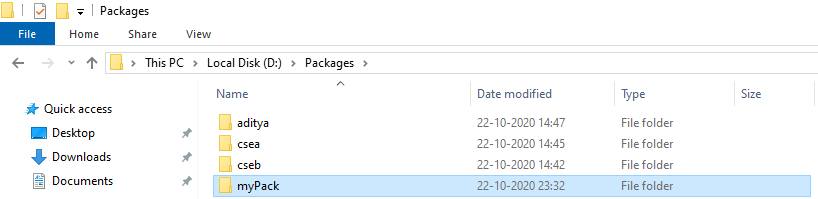
**Step – 1:**

****

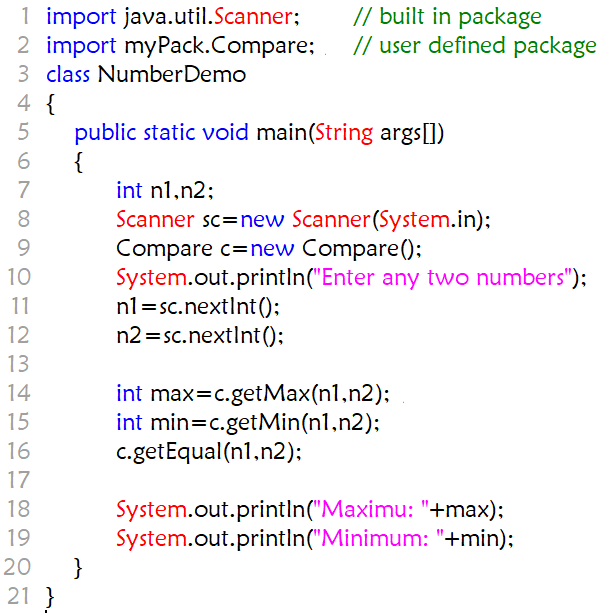
****

**Step – 2:**

****

****

**Step – 3 & 4:**

****

**Output:**

Enter any two numbers

5

4

Not Equal

Maximum: 5

Minimum: 4

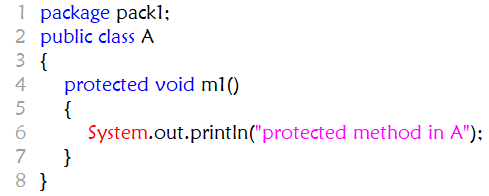
**8.2) Write a Java program to illustrate the use of protected members in a package.**

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

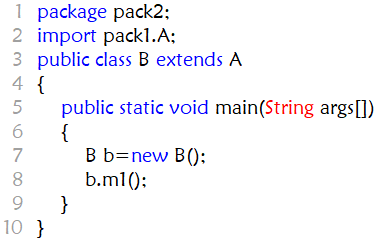
The protected access modifier can be applied on the data member, method and constructor. It can't be applied on the class.

**Code:**

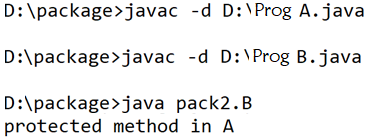
**CASE – 1:**



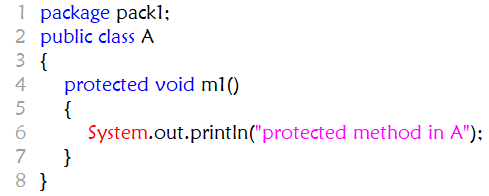
=========================================



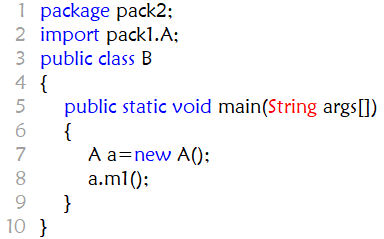
**Output:**



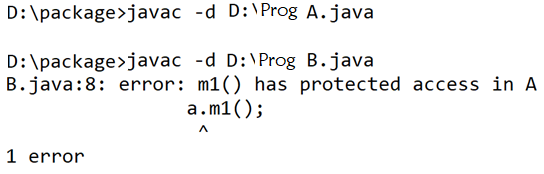
**CASE – 2:**



=============================================



**Output:**



**Experiment-9: Exception Handling**

**9.1) Write a Java program to illustrate exception handling mechanism using multiple catch clauses.**

**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);

}

}

}

**Output:**

E:\T-Hub>javac ExceptionDemo.java

E:\T-Hub>java ExceptionDemo 1 0

/ by zero

Cleanup code

0

**9.2) Write a Java program to make use of Built-in and user-defined Exceptions in handling a run time exception.**

**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).

**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:**

E:\T-Hub>javac Excep3.java

E:\T-Hub>java Excep3 -100 100 -200 200

Marks should be greater than 0

**Experiment-10: Multithreading**

**10.1) 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).**

**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);

}

}

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:**

MAIN CLOSED

B: Hello

A: Good Morning

C: Welcome

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

**10.2) Write a Java program to solve Producer-Consumer problem using synchronization.**

**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;

}

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);

}

}

}

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:**

Enter data

CSE

Produced:CSE

Consumed:CSE

Enter data

IT

Produced:IT

Consumed:IT

Enter data

ECE

Produced:ECE

Consumed:ECE

Enter data

EEE

Produced:EEE

Consumed:EEE

Enter data

AIML

Produced:AIML

Consumed:AIML

Enter data

**Experiment-11: Event Handling**

**11.1) Write a Java program to illustrate the Keyboard Events by using an applet code.**

**Code:**

import java.applet.Applet;

import java.awt.\*;

import java.awt.event.\*;

/\* <APPLET CODE ="KeyEvents.class" WIDTH=300 HEIGHT=200> </APPLET> \*/

public class KeyEvents extends Applet implements KeyListener {

TextField t,tt,tp,tr;

public void init() {

t=new TextField(20);

t.addKeyListener(this);

tt=new TextField(70);

tp=new TextField(70);

tr=new TextField(70);

add(t);

add(tt);

add(tr);

add(tp);

}

public void keyTyped(KeyEvent e) {

tt.setText("key Typed: "+e.getKeyChar());

}

public void keyReleased(KeyEvent e) {

tr.setText("key Released: "+e.getKeyChar());

}

public void keyPressed(KeyEvent e) {

int kc;

String s;

kc=e.getKeyCode();

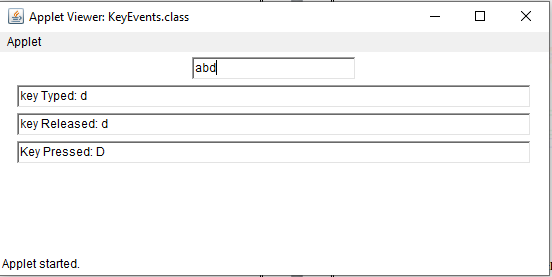
s=e.getKeyText(kc);

tp.setText("Key Pressed: "+s);

}

}

**Output:**

****

**11.2) Write a Java program to illustrate the Mouse Events by using an applet code.**

**Code:**

import java.awt.\*;

import java.awt.event.\*;

class MouseListenerExample extends Frame implements MouseListener{

Label l;

MouseListenerExample(){

addMouseListener(this);

l=new Label();

l.setBounds(20,50,100,20);

add(l);

setSize(300,300);

setLayout(null);

setVisible(true);

}

public void mouseClicked(MouseEvent e) {

l.setText("Mouse Clicked");

}

public void mouseEntered(MouseEvent e) {

l.setText("Mouse Entered");

}

public void mouseExited(MouseEvent e) {

l.setText("Mouse Exited");

}

public void mousePressed(MouseEvent e) {

l.setText("Mouse Pressed");

}

public void mouseReleased(MouseEvent e) {

l.setText("Mouse Released");

}

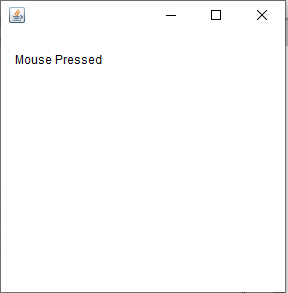
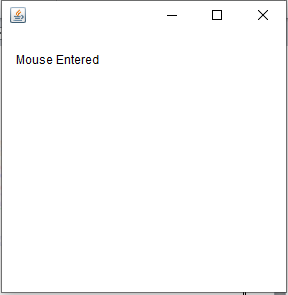
public static void main(String[] args) {

new MouseListenerExample();

}

}

**Output:**

**** ****

**11.3) Write a Java program to generate a simple calculator using AWT components.**

**Code:**

import java.awt.\*;

import java.awt.event.\*;

class Calc extends Frame implements ActionListener{

TextField tf1,tf2,tf3;

Button b1;

public Calc(){

setLayout(new FlowLayout());

tf1=new TextField(20);

tf2=new TextField(20);

tf3=new TextField(20);

b1=new Button("ADD");

add(tf1);

add(tf2);

add(tf3);

add(b1);

b1.addActionListener(this);

}

public void actionPerformed(ActionEvent ae){

if(ae.getSource()==b1){

int a,b,c;

a=Integer.parseInt(tf1.getText());

b=Integer.parseInt(tf2.getText());

c=a+b;

tf3.setText(c+"");

}

}

public static void main(String args[]){

Calc c=new Calc();

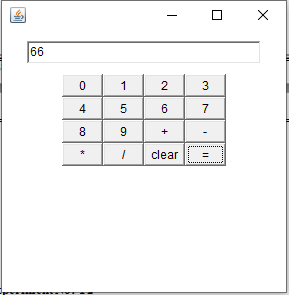
c.setSize(400,400);

c.setVisible(true);

}

}

**Output:**



**Experiment-12: Write a JDBC program to perform the following operations by connecting to MYSQL database.**

**Table Creation:**

import java.sql.\*;//import the sql package to perform the database connectivity

public class TableCreation

{

public static void main(String args[])throws Exception

{

Class.forName("com.mysql.cj.jdbc.Driver");

Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/politics","root", "0000");

if(con!=null)

System.out.println("Database connected");

Statement st=con.createStatement();

String query="create table Assembly(areacode varchar(10),mla varchar(20),population integer(6))";

//as we used create statement in the above line, create can only be used in executeUpdate();

st.executeUpdate(query);

System.out.println("Query executed successfully!");

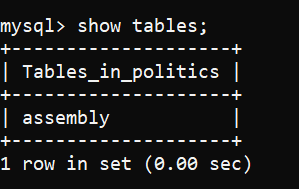
}

}

**Output:**

Database connected

Query executed successfully!



**12.1) Inserting Data into Table.**

**Code:**

import java.sql.\*;//import the sql package to perform the database connectivity

public class InsertionDemo

{

public static void main(String args[])throws Exception

{

Class.forName("com.mysql.cj.jdbc.Driver");

Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/politics","root", "0000");

if(con!=null)

System.out.println("Database connected");

Statement st=con.createStatement();

String query="insert into Assembly values('EG1301','Chinna Rajappa',600000)";

st.executeUpdate(query);

query="insert into Assembly values('WG1302','RRR',400000)";

st.executeUpdate(query);

query="insert into Assembly values('GN1303','Kodali Nani',700000)";

st.executeUpdate(query);

System.out.println("Query executed successfully!");

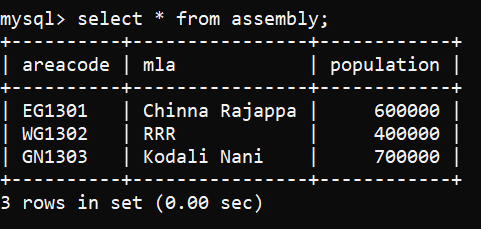
}

}

**Output:**

Database connected

Query executed successfully!



**12.2) Updating Data in the Table.**

**Code:**

import java.sql.\*;//import the sql package to perform the database connectivity

public class UpdationDemo

{

public static void main(String args[])throws Exception

{

Class.forName("com.mysql.cj.jdbc.Driver");

Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/politics","root", "0000");

if(con!=null)

System.out.println("Database connected");

Statement st=con.createStatement();

String query="update Assembly set population=450000 where areacode='EG1301'";

st.executeUpdate(query);

System.out.println("Query executed successfully!");

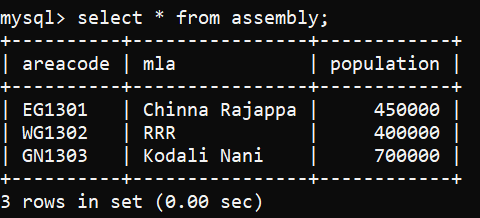
}

}

**Output:**

Database connected

Query executed successfully!



**12.3) Deleting Data from the Table based on a column value.**

**Code:**

import java.sql.\*;//import the sql package to perform the database connectivity

public class DeletionDemo

{

public static void main(String args[])throws Exception

{

Class.forName("com.mysql.cj.jdbc.Driver");

Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/politics","root", "0000");

if(con!=null)

System.out.println("Database connected");

Statement st=con.createStatement();

String query="delete from Assembly where areacode='EG1301'";

st.executeUpdate(query);

System.out.println("Query executed successfully!");

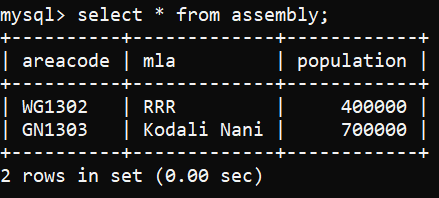
}

}

**Output:**

Database connected

Query executed successfully!



**AUGMENTED EXPERIMENTS:**

**Experiment-14:** **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 colour.**

**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");

}

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:**

**** ****

**Experiment-16:** **Develop an applet 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” is clicked.**

**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);

}

}

**Output:**

