**Experiment No.1**

**Aim:** **Programs on Operators, Arithmetic Promotion, Method Calling.**

1. Implement a Java Program to determine largest of 3 numbers as long as user wants.

Program :

import java.util.\*;

import java.io.\*;

public class LargestNo {

public static void main(String args[]) {

int ch, max , a, b, c;

PrintWriter pw = new PrintWriter(System.out,true);

Scanner sc = new Scanner(System.in);

do {

pw.println(" Enter 3 numbers ");

a = sc.nextInt();

b = sc.nextInt();

c = sc.nextInt();

max = ( a > b) ? ( a > c ? a : c ) : ( b > c ? b : c);

pw.println(" Largest number among 3 numbers : "+ max);

pw.println(" Do you want to continue \n 1.Yes \t 0.No");

ch = sc.nextInt();

}while( ch != 0 );

}

}

1. Implement a Java Program to demonstrate method calling to find prime numbers from the range of given numbers.

Program : import java.util.\*;

public class Prime {

public void prime(int start, int end) {

for( int i = start; i <= end; i++) {

int flag= 0;

for(int j = 2; j <= (i/2); j++) {

if( i % j == 0 ) {

flag = 1;

break ;

}

}

if( flag != 1 )

System.out.println(i);

}

}

public static void main(String args[]) {

Scanner sc = new Scanner(System.in);

Prime obj = new Prime();

System.out.println("Program to find prime between given range");

System.out.println("Enter start of range") ;

int start = sc.nextInt();

System.out.println("Enter end of range") ;

int end = sc.nextInt();

obj.prime(start,end);

}

}

**Experiment No.2**

**Aim:** Programs on dealing with Arrays.

1. Implement a Java Program to sort elements of an array in an ascending and descending order using 1D array.

Program :

import java.util.\*;

class sort{

int i,j,temp,n;

int a[]=new int[20];

void getdata() {

Scanner sc=new Scanner(System.in);

System.out.println("Enter no of elements in array");

n=sc.nextInt();

System.out.println("Enter the numbers:");

for(i=0;i<n;i++) {

a[i]=sc.nextInt();

}

}

void asc() {

for(i=0;i<n;i++) {

for(j=i+1;j<n;j++) {

if(a[i]>a[j]) {

temp=a[i];

a[i]=a[j];

a[j]=temp;

}

}

}

System.out.println("Ascending order:");

for(i=0;i<n;i++)

System.out.println(a[i]);

}

void desc() {

for(i=0;i<n;i++) {

for(j=i+1;j<n;j++) {

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

temp=a[i];

a[i]=a[j];

a[j]=temp;

}

}

}

System.out.println("Descending order:");

for(i=0;i<n;i++)

System.out.println(a[i]);

}

}

class exp\_2\_1{

public static void main(String art[]) {

sort s=new sort();

s.getdata();

s.asc();

s.desc();

}

}

1. Implement a Java Program to perform multiplication of two matrices using 2D array.

Program :

import java.util.\*;

class exp\_2\_2{

public static void main(String ar[]){

int n;

Scanner sc=new Scanner(System.in);

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

n=sc.nextInt();

int a[][] = new int[n][n];

int b[][] = new int[n][n];

int c[][] = new int[n][n];

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

for(int i=0;i<n;i++){

for(int j=0;j<n;j++) {

a[i][j]=sc.nextInt();

}

}

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

for(int i=0;i<n;i++){

for(int j=0;j<n;j++){

b[i][j]=sc.nextInt();

}

}

System.out.println("Multiplying the matrices");

for(int i=0;i<n;i++){

for(int j=0;j<n;j++){

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

c[i][j]=c[i][j]+a[i][k]\*b[k][j];

}

}

}

System.out.println("The product is:");

for(int i=0;i<n;i++){

for(int j=0;j<n;j++){

System.out.println(c[i][j]+"");

}

System.out.println();

}

sc.close();

}

}

**Experiment No.3**

**Aim:** Programs on Classes: String and Math.

1. Implement a Java Program to demonstrate at least 10 String methods and Implement a program that computes your initials from your full name and displays them.

Program :

import java.lang.String;

public class StringDemo {

public static void main(String args[]) {

String str1 = "aparna";

String str2 = "aparna";

String str3 = new String("aparna");

String str4 = new String("APARNA");

System.out.println("str1:"+str1.hashCode()+"\nstr2:"+str2.hashCode()+"\nstr3 :"+str3.hashCode()+"\nstr4 :"+str4.hashCode());

System.out.println("char at index 2"+str1.charAt(2));

System.out.println("equals of string :"+str1.equals(str4));

System.out.println("equals of string ignoring the case :"+str1.equalsIgnoreCase(str4));

System.out.println("compare 2 str : "+str1.compareTo(str4));

System.out.println("check presence of letters :"+str1.contains("apa"));

System.out.println("subString :"+str1.substring(2));

System.out.println("substring :"+str1.substring(0,2));

System.out.println("index of perticular char a : "+str1.indexOf('a'));

System.out.println("last index of perticular char : "+str1.lastIndexOf('a'));

System.out.println("check wether string is empty or not : "+str1.isEmpty());

System.out.println("replace a with $ : "+str1.replace('a','$'));

System.out.println("length : "+str1.length());

System.out.println("to upper case : "+str1.toUpperCase());

System.out.println("to lower : "+str4.toLowerCase());

}

}

1. Implement a Java Program to demonstrate at least 10 math class methods and Implement a program using switch...case statement to determine the amount of electricity bill based on following logic:

1-100 units – Rs.6.0

100-200 units – Rs.7.00

200-500 units – Rs.8.00

more than 500 – Rs.9.00

Program :

import java.util.\*;

public class ElectricBill {

public static void main(String args[]) {

System.out.println("Enter light units");

Scanner sc = new Scanner(System.in);

int units = sc.nextInt();

double bill = 0.0;

if( units > 0 ) {

if( units <= 100) {

bill = units \* 6;

}

else {

bill = bill + ( 100 \* 6 );

units = units - 100;

int ch = 1;

while( units > 0 ) {

switch(ch) {

case 1: // System.out.println("case 1"+"\n bill : "+bill+"\nunits :"+units);

if( units <= 200 ) bill = bill + ( 7 \* units);

else bill = bill + (7 \* 200);

units = units - 200;

ch = 2;

break;

case 2: System.out.println("case 2");

if( units <= 500 ) bill = bill + ( 8 \* units);

else bill = bill + (8 \* 500);

units = units - 500;

break;

default : System.out.println("default");

bill = bill + (9 \* units);

units = 0;

break;

}//end of switch

}//end of while

}//end of else

System.out.println("bill : "+ bill);

}//emd of outer if

else System.out.println("invalid units");

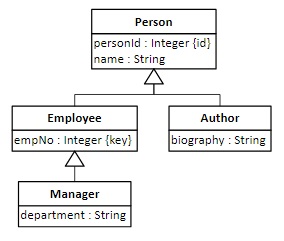
}

}

**Experiment No.4**

**Aim:** Programs on Inheritance and Polymorphism.

1. Implement a Java program to demonstrate Multilevel and Hierarchical inheritance for given example. Define the required constructors and methods for classes.



Program :

import java.util.\*;

class Person {

int personId;

String name;

Person(int id , String name) {

this.name = name;

this.personId = id;

}

void display()

{

System.out.println("Person details:");

System.out.println("Person id:"+personId);

System.out.println("Person name:"+name);

}

}

class Emp extends Person{

Emp(int id, string name, int empno) {

super(id,name);

this.empno=empno;

}

void display()

{

super.display();

System.out.println("\nEmp details:");

System.out.println("\nEmp no:"+empno);

}

}

class Author extends Person{

String biography;

Author(int id, String name, int emp\_no, String bio ) {

super(id,name,emp\_no);

biography=bio;

}

void display()

{

super.display();

System.out.println("\nAuthor details:");

System.out.println("Author biography:"+biography);

}

}

class Manager extends Emp{

String department;

Manager(int id, String name, int emp\_no, String dept) {

super(id,name,emp\_no);

department=dept;

}

void display()

{

super.display();

System.out.println("\nManager details:");

System.out.println("\nManager departent:");

}

}

public class InheritanceDemo

{

public static void main(String args[])

{ Scanner sc = new Scanner(System.in);

int id, empno;

String name, dept, bio;

System.out.println("\n Enter person id, name, emp\_no, department for manager");

id = sc.nextInt();

name = sc.next();

empno = sc.nextInt();

dept = sc.next();

Manager m=new Manager(id,name,empno,dept);

System.out.println("\n Enter person id, name, emp\_no, biography for Author");

id = sc.nextInt();

name = sc.next();

empno = sc.nextInt();

bio = sc.next();

Author B=new Author(id,name,empno,bio);

m.display();

B.display();

}

}

1. Implement a Java program to find area of Square, Rectangle and Circle using Method Overloading.

Program :

import java.util.\*;

class Calareas

{

double CirArea,RectArea,SquArea;

double side;

double area(double s)

{

side=s;

SquArea=side\*side;

return SquArea;

}

double area(int rad)

{

CirArea=3.14\*(rad\*rad);

return CirArea;

}

double area(int l,int b)

{

RectArea=l\*b;

return RectArea;

}

public static void main(String[] args)

{

Calareas c=new Calareas();

Scanner sc=new Scanner(System.in);

double a1,a2,a3;

int c\_radius,r\_l,r\_b;

double s\_s;int ch;

do

{

System.out.println("\n1.Square \n 2.Circle \n 3.Rectangle \n4.exit \nEnter your choice:");

ch=sc.nextInt();

switch(ch)

{

case 1:

System.out.println("\n# Square:");

System.out.println("Enter the value of side=");

s\_s=sc.nextDouble();

a1=c.area(s\_s);

System.out.println("Area of Square ="+a1);

break;

case 2:

System.out.println("\n# Circle:");

System.out.println("Enter the value of radius=");

c\_radius=sc.nextInt();

a2=c.area(c\_radius);

System.out.println("Area of Circle ="+a2);

break;

case 3:

System.out.println("\n# Rectangle:");

System.out.println("Enter the value of l and b=");

r\_l=sc.nextInt();

r\_b=sc.nextInt();

a3=c.area(r\_l,r\_b);

System.out.println("Area of Rectangle ="+a3);

break;

}

}while(ch!=4);

}

}

**Experiment No.5**

**Aim:** Programs on Garbage collection, packaging, access Modifiers, as well as static and abstract modifiers.

1. Implement a Java program to create user defined package arithmetic package to perform basic arithmetic functionalities such as addition, subtraction, multiplication & division.

Program :

1. Implement a Java program to demonstrate static and abstract modifiers.

Program :

// Creating Package

package operations;

public class Cal {

public int add(int a, int b) {

return (a+b);

}

public int sub(int a, int b) {

return (a-b);

}

public int div(int a, int b) {

return (a/b);

}

public int mul(int a, int b) {

return (a\*b);

}

}

// Importing Package

import operations.Cal;

import java.util.\*;

public class PackageDemo {

public static void main(String args[]) {

Scanner sc = new Scanner(System.in);

System.out.println("\n Enter two values for calculations : \t ");

int no1 = sc.nextInt();

int no2 = sc.nextInt();

Cal obj = new Cal();

System.out.println(" Addition : "+obj. add(no1,no2));

}

}

**Experiment No.6**

**Aim:** **Implement a Java program to demonstrate constructor overloading for Date Class containing three attributes such as day, month and year and following constructors**

i) No-argument constructor: initializes day, month and year to default values 1, January & 1970 respectively.

ii) Constructor with one parameter: day supplied, month and year defaulted to January & 1970 respectively.

iii) Constructor with two parameters: day and month are supplied, year defaulted to 1970.

iv) Constructor with three parameters: date, month and year are supplied.

Program :

import java.util.\*;

public class ContructorOverloading {

int day;

int month;

int year;

int[] DpM = {0,31,28,31,30,31,30,31,31,30,31,30,31};

ContructorOverloading() {

day = 1;

month = 1;

year = 1970;

}

ContructorOverloading(int d) {

month = 1;

year = 1970;

//checking the vaidity of day

if( d <= 0 || d > DpM[month]) throw new IllegalArgumentException("day out of range");

day = d;

}

}

import java.util.\*;

public class Date {

public static void main(String args[]) {

try {

ContructorOverloading date1 = new ContructorOverloading();

System.out.println("\n day "+ date1.day);

ContructorOverloading date2 = new ContructorOverloading(56);

System.out.println("\n day "+ date1.day);

}

catch(IllegalArgumentException e) {

System.out.println(e);

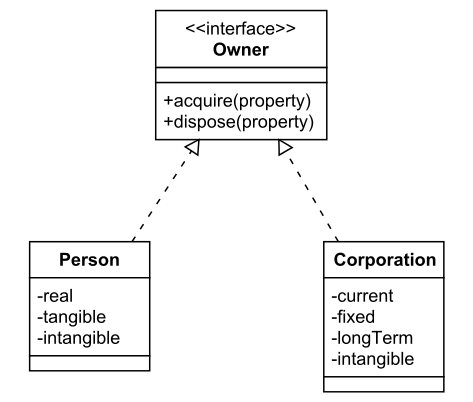
}

} }

**Experiment No.7**

**Aim: Program on Interfaces block initializers, final Modifier, as well as static and dynamic binding.**

a. Implement a Java program to demonstrate interfaces in java with the following example



interface Owner {

void aquier(int a);

void dispose(int a);

}

class Person implements Owner {

public static int real = 10;

public static int tangible = 15;

public static int intangible = 20;

public void aquier(int a){

if( a == real ) real++;

if( a == tangible ) tangible ++;

else intangible++;

}

public void dispose(int a) {

if( a == real ) real--;

if( a == tangible ) tangible --;

else intangible--;

}

}

class Corporation implements Owner {

public static int current = 10;

public static int fixed = 15;

public static int intangible = 25;

public static int longTerm = 25;

public void aquier(int a){

if( a == current ) current++;

if( a == fixed ) fixed ++;

if( a == longTerm ) longTerm ++;

else intangible++;

}

public void dispose(int a) {

if( a == current ) current--;

if( a == fixed ) fixed --;

if( a == longTerm ) longTerm --;

else intangible--;

}

}

public class InterfaceDemo {

public static void main(String args[]) {

Person p = new Person();

p.aquier(Person.real);

System.out.println(Person.real);

p.aquier(Person.tangible);

System.out.println(Person.tangible);

p.aquier(Person.intangible);

System.out.println(Person.intangible);

p.dispose(Person.real);

System.out.println(Person.real);

p.dispose(Person.tangible);

System.out.println(Person.tangible);

p.dispose(Person.intangible);

System.out.println(Person.intangible); } }

**Experiment No.8**

**Aim: Programs on file handling and stream manipulation.**

1. Implement a Java Program to find the sum of all the primes between given range of integers. Read the required data using methods of DataInputStream class.

Program :

import java.io.\*;

//import java.util.\*;

public class PrimeNo {

public static int prime(int start, int end) throws IOException{

int sum = 0;

for( int i = start; i <= end; i++) {

int flag= 0;

for(int j = 2; j <= (i/2); j++) {

if( i % j == 0 ) {

flag = 1;

break ;

}

}

if( flag != 1 ) {

//System.out.println(i);

sum +=i;

}

}

return sum;

}

public static void main(String args[]) {

try {

PrintWriter pw = new PrintWriter(System.out,true);

DataInputStream in = new DataInputStream(System.in);

//Scanner sc = new Scanner(System.in);

pw.println("Program to find prime between given range");

pw.println("Enter start of range") ;

int start = Integer.parseInt(in.readLine());

pw.println("Enter end of range") ;

int end = Integer.parseInt(in.readLine());

pw.println( prime(start,end));

}

catch(IOException e) { }

}

}

1. Implement a Java Program to copy the contents of source text file to destination file and also print the number of characters, words and lines in source file using FileInputStream and FileOutputStream class.

Program :

import java.io.\*;

public class FileDemo {

public static void main(String args[ ]) throws IOException{

File inputFile = new File("F:\\java\_programs\\inputFile.txt");

File outputFile = new File("F:\\java\_programs\\outputFile.txt");

FileInputStream in =new FileInputStream(inputFile);

FileOutputStream out =new FileOutputStream(outputFile);

int i , c = 0, s = 0 , l =0 ;

char ch;

while( ( i = in.read() ) != -1 ) {

out.write(i);

ch = (char)i;

System.out.print(ch);

if( ch == ' ') s++;

if( ch == '\n') l++;

else c++;

}

System.out.println("characters : "+ c+"\t lines : "+ (l+1) +"\t spaces : "+s);

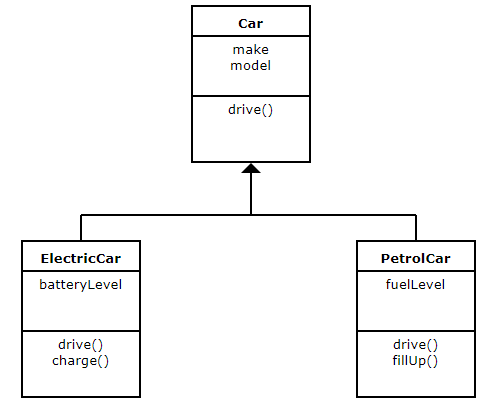
}

}

**Experiment No.9**

**Aim:** **Program on Dynamic Polymorphism.**

1. **Implement a Java program to demonstrate method overriding for following example**



Program :

class Car {

int mke;

String model;

public void drive() {

System.out.println(" car's drive ");

}

}

class ElectricCar extends Car {

float batteryLevel;

public void drive() {

System.out.println(" ElectricCar's drive ");

batteryLevel--;

}

public void charge() {

batteryLevel++;

}

}

class PetrolCar extends Car {

int FuelLevel;

public void drive() {

System.out.println(" PetrolCar's drive ");

fuelLevel--;

}

public void fillup() {

fuelLevel++;

}

}

public class OverridenCar {

public static void main(String args[]) {

Car c = new Car();

ElectricCar ec = new ElectricCar();

PetrolCar pc = new PetrolCar();

Car t;

t = c;

t.drive();

t = ec;

t.drive();

t = pc;

t.drive();

}

}

**Experiment No.10**

**Aim: Program on Exception Handling.**

1. Implement a Java program to demonstrate exception handling using try-catch, throw, throws, finally blocks.

Program :

class ExceptionDemo {

public void demo() throws Arithmatic Exception {

int a =10, b = 0, c;

c = a / b;

System.out.println(" this will not display");

}

public static void main(String args[]) {

try {

demo();

throw new NullPointerEXception();

}

catch(Arithmatic Exception e) { }

catch(NullPointerEXception e) { }

finally {

System.out.println("this is finally"); } } }

**Experiment No.11**

**Aim: Program on generic programming in java.**

1. Implement a Java Program to print an array of different type using a single Generic method i.e. printArray. Create objects of Integer, Double and Character classes with 5 elements each and sort those using Arrays class methods. To display the array contents, call Generic method printArray.

Program :

import java.util.\*;

import java.io.\*;

public class GenericDemo {

public static void main(String args[]) {

Scanner sc = new Scanner(System.in); // to take input

PrintWriter pr = new PrintWriter(System.out,true); // to print on console

// declaring arrays of wrapup classes

Integer[] in = { 5,1,3,4,2};

Double[] db = new Double[5];

Character[] ch = new Character[5];

// innitializing arrays

pr.println("\n Enter 5 double values");

for( int i = 0; i < 5; i++)

db[i] = sc.nextDouble();

pr.println("\n Enter 5 characters");

for( int i = 0; i < 5; i++)

ch[i] = sc.next().charAt(0);

// Sorting elements in arrays

Arrays.sort(in);

Arrays.sort(db);

Arrays.sort(ch);

//print elements of array

printArray(in);

printArray(db);

printArray(ch);

}

public static <T> void printArray( T[] inputele) {

for( T ele : inputele )

System.out.printf(" %s",ele);

System.out.println();

}

}

**Experiment No.12**

**Aim: Program on iterators in Java.**

1. Implement a Java Program to demonstrate use of iterators. Create an array list of integers with ArrayList class and display the contents of list using Iterator object.

Program :

import java.util.\*;

import java.io.\*;

public class IteratorDemo {

public static void main(String args[]) {

ArrayList<Integer> list = new ArrayList<Integer>();

for(int i = 1; i < = 5; i++)

list.add(i);

Iterator itr = list.iterator();

while(itr.hasNext())

System.out.println(itr.next()); } }