|  |
| --- |
|  |
| LAB EXPERIMENTS |
|  |
|  |
| **Vansh Sukhija**  **12112021** |
|  |

|  |
| --- |
|  |

Experiment 1

Ans 1-

import java.lang.\*;

import java.util.\*;

class Alias{

*float* flt = 0.69f;

}

public class aliasing1{

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

*Alias* al = new Alias();

*Alias* temp = al;

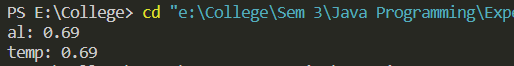
        System.out.println("al: "+al.flt);

        System.out.println("temp: "+temp.flt);

    }

}

Output-



Ans 2-

import java.lang.\*;

import java.util.\*;

public class prime2{

    public static *void* main(*String* *agrs*[]){

*Scanner* scan = new Scanner(System.in);

        System.out.print("Enter a number: ");

*int* num = scan.nextInt();

        System.out.print("Prime numbers till "+num+" are: ");

        for(*int* i=2; i<=num; i++){

*int* j;

            for(j=2; j<i; j++){

                if(i%j==0)

                    break;

            }

            if(j==i)

                System.out.print(i+" ");

        }

        System.out.println();

    }

}

Output-



Ans 3-

import java.lang.\*;

import java.util.\*;

public class switch3{

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

        for(*int* i=0; i<7; i++){

            switch(i){

                case 0:

                    System.out.println("Sunday");

                    break;

                case 1:

                    System.out.println("Monday");

                    break;

                case 2:

                    System.out.println("Tuesday");

                    break;

                case 3:

                    System.out.println("Wednesday");

                    break;

                case 4:

                    System.out.println("Thursday");

                    break;

                case 5:

                    System.out.println("Friday");

                    break;

                case 6:

                    System.out.println("Saturday");

                    break;

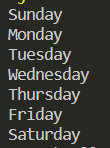
            }

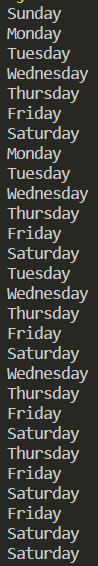
        }

    }

}

Output-





Ans 4-

import java.lang.\*;

import java.util.\*;

class Reference{

*String* str = new String();

}

public class reference4{

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

*Reference* ref = new Reference();

        System.out.println("String: "+ref.str);

    }

}

Output-



Experiment 2

Ans 1-

import java.lang.\*;

import java.util.\*;

public class comparison1{

    static *void* compare(*String* *str1*, *String* *str2*){

        if(str1.equals(str2))

            System.out.println("Both strings are equal");

        else

            System.out.println("Strings are not equal");

    }

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

*Scanner* scan = new Scanner(System.in);

*String* str1, str2;

        System.out.print("Enter 1st String: ");

        str1 = scan.nextLine();

        System.out.print("Enter 2nd String: ");

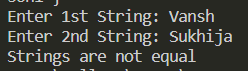
        str2 = scan.nextLine();

        compare(str1, str2);

    }

}

Output-



Ans 2-

import java.lang.\*;

import java.util.\*;

class Dog{

*String* name, says;

    public *void* display(){

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

        System.out.println("Says: "+says);

        System.out.println();

    }

};

public class ans2 {

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

*Dog* Spot = new Dog();

*Dog* Scruffy = new Dog();

        Spot.name = "Spot";

        Spot.says = "Ruff!";

        Scruffy.name = "Scruffy";

        Scruffy.says = "Wurf!";

        Spot.display();

        Scruffy.display();

*Dog* Temp = Spot;

        if(Temp==Spot)

            System.out.println("Temp and Spot are equal");

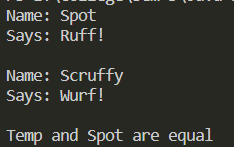
        else

            System.out.println("Temp and Spot are not equal");

    }

}

Output-



Experiment 4

Ans 1-

import java.util.\*;

public class ans1 {

    static String str2;

    static{

        str2 = "Str-2";

    }

    static class Temp{

        static String str1 = "Str-1";

        static *void* print(){

            System.out.println(str1);

            System.out.println(str2);

        }

    }

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

        Temp t = new Temp();

        t.print();

    }

}

Output-



Ans 2-

import java.util.\*;

public class ans2 {

    public static *void* main(String... *args*){

        for(*int* i=0; i<*args*.length; i++){

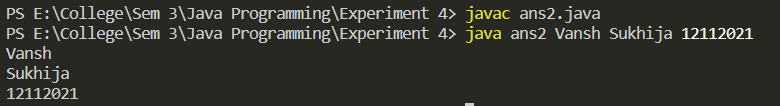
            System.out.println(*args*[i]);

        }

    }

}

Output-



|  |
| --- |
|  |

|  |
| --- |
|  |

Experiment 5

Ans 1-

import java.util.\*;

public class ans1 {

    static class A{

        A(){

            System.out.println("Constructor for A");

        }

    };

    static class B{

        B(){

            System.out.println("Constructor for B");

        }

    };

    static class C extends A{

        B temp;

    }

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

        C temp = new C();

    }

}

Output-



Ans 2-

import java.util.\*;

public class ans2 {

    static class base{

        base(*int* *a*){

            System.out.println("Base is constructed");

        }

    };

    static class derived extends base{

        derived(){

            super(10);

            System.out.println("Default constructor of Derived");

        }

        derived(*int* *a*){

            super(*a*);

            System.out.println("Non-Default constructor of Derived");

        }

    };

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

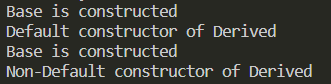
        derived d = new derived();

        derived de = new derived(69);

    }

}

Output-



Ans 3-

import java.util.\*;

public class ans3 {

    static class base{

*void* meth(){

            System.out.println("Method 1");

        }

*void* meth(*int* *a*){

            System.out.println("Method 2");

        }

*void* meth(*int* *a*, *int* *b*){

            System.out.println("Method 3");

        }

    };

    static class derived extends base{

*void* meth(*double* *a*){

            System.out.println("Method 4");

        }

    };

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

        derived d = new derived();

        d.meth();

        d.meth(10);

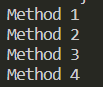
        d.meth(10, 20);

        d.meth(0.69);

    }

}

Output-



Ans 4-

import java.util.\*;

public class ans4 {

    static class A{

        final *void* printf(){

            System.out.println("printf() of class A");

        }

    }

    static class B extends A{

*void* printf(){

            System.out.println("printf() of class B");

        }

    }

    static final class C{

*void* temp(){

            System.out.println("Method of final class C");

        }

    }

    static class D extends C{

*void* doratheexplorer(){

            System.out.println("Method of class D inherited by C");

        }

    }

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

        A a = new A();

        a.printf();

        B b = new B();

        b.printf();

        C c = new C();

        c.temp();

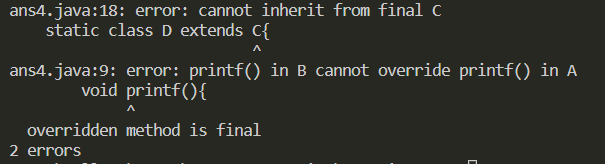
        D d = new D();

        d.doratheexplorer();

    }

}

Output-



Experiment 6

Ans 1-

class Temp{

    private *void* pri\_method(){

        System.out.println("This is a Private method");

    }

    public *void* pub\_method(){

        System.out.println("This is a Public method");

    }

    protected *void* pro\_method(){

        System.out.println("This is a Protected method");

    }

};

public class ans1 {

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

        Temp t = new Temp();

        t.pri\_method();

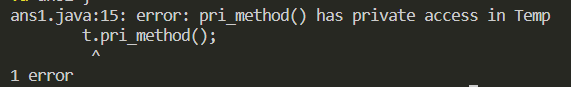
        t.pub\_method();

        t.pro\_method();

    }

}

Output-





Experiment 7

Ans 1-

public class ans1 {

    static abstract class base{

        abstract *void* print();

        base(){

            print();

        }

    }

    static class derived extends base{

*int* num = 69;

*void* print(){

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

        }

    }

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

        derived d = new derived();

        d.print();

    }

}

Output-



Ans 2-

public class ans2 {

    interface tempinterface{

*void* method1();

    }

    static class tempclass implements tempinterface{

        public *void* method1(){

            System.out.println("Public method 1");

        }

    }

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

        tempclass t = new tempclass();

        t.method1();

    }

}

Output-



Ans 3-

public class ans3 {

    interface A{

*void* A1();

*void* A2();

    }

    interface B{

*void* B1();

*void* B2();

    }

    interface C{

*void* C1();

*void* C2();

    }

    interface D extends A, B, C{

*void* D1();

    }

    static class temp implements D{

        public *void* A1(){

            System.out.println("Function A1");

        }

        public *void* A2(){

            System.out.println("Function A2");

        }

        public *void* B1(){

            System.out.println("Function B1");

        }

        public *void* B2(){

            System.out.println("Function B2");

        }

        public *void* C1(){

            System.out.println("Function C1");

        }

        public *void* C2(){

            System.out.println("Function C2");

        }

        public *void* D1(){

            System.out.println("Function D1");

        }

    }

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

        temp t = new temp();

        t.A1();

        t.B1();

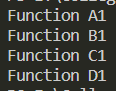
        t.C1();

        t.D1();

    }

}

Output-



Experiment 8

Ans 1-

public class ans1 {

    static class outer{

        private *int* temp=10;

        private *void* outermeth1(){

            System.out.println("Value of temp: "+temp);

        }

        public class inner{

            public *void* innermeth1(){

                temp++;

                outermeth1();

            }

        }

        public *void* outermeth2(){

            inner i = new inner();

            i.innermeth1();

        }

    }

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

        outer o = new outer();

        o.outermeth2();

    }

}

Output-



Ans 2-

public class ans2 {

    static class outer{

        public *void* outermeth(){

            inner i = new inner();

            i.innermeth();

        }

        private class inner{

            private *void* innermeth(){

                System.out.println("This is the private method of the inner class");

            }

        }

    }

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

        outer o = new outer();

        o.outermeth();

    }

}

Output-



Experiment 9

Ans 1-

import java.io.FileNotFoundException;

public class ans1 {

    ans1(String *msg*){

        System.out.println("Message in Constructor");

    }

    public static *void* main(String[] *args*)

            throws Exception {

        try {

            throw new FileNotFoundException();

        }

        catch (FileNotFoundException e) {

            throw new Exception("File not found");

        }

        catch (Exception e) {

            System.out.println(e.getMessage());

        }

        finally {

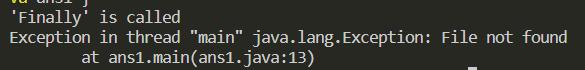
            System.out.println("'Finally' is called");

        }

    }

}

Output-



Ans 2-

class UserException extends Exception {

    UserException(String *msg*) {

        super(*msg*);

    }

}

class ExceptionTesting {

*void* display(*int* *x*) throws UserException {

        System.out.println("Value of x is: " + *x*);

        throw new UserException("Exception Occured");

    }

}

public class ans2 {

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

        ExceptionTesting ob1 = new ExceptionTesting();

        try {

            ob1.display(50);

        }

        catch (UserException e) {

            System.out.println("Error found: " + e.getMessage());

        }

    }

}

Output-

