1. Write a program in java to implement single level inheritance and display the visibility of different access specifiers (private, public, protected).

class parent {

    void output1() {

        System.out.println("Inside default access specifier.");

    }

    public void output2() {

        System.out.println("Inside public access specifier.");

    }

    private void output3() {

        System.out.println("Inside private access specifier.");

    }

    protected void output4() {

        System.out.println("Inside protected access specifier.");

    }

    public static void main(String[] args) {

        parent par = new parent();

        par.output1();

        par.output2();

        par.output3();

        par.output4();

    }

}

class derived extends parent {

    derived() {

        super.output1();

        super.output2();

        // super.output3(); // private method can't be accessed from derived class.

        super.output4();

    }

}

class another {

    another() {

        parent p = new parent();

        p.output1();

        p.output2();

        // p.output3(); // private method can't be accessed outside the class.

        p.output4();

    }

}

class checkAccess {

    public static void main(String[] args) {

        derived d = new derived();

        another obj = new another();

    }

}

OUTPUT:

Microsoft Windows [Version 10.0.19045.3324]

(c) Microsoft Corporation. All rights reserved.

C:\Users\Anurag Singh>d:

D:\>cd desktop

D:\Desktop>cd java

D:\Desktop\Java>javac fourteen.java

D:\Desktop\Java>java checkAccess

Inside default access specifier.

Inside public access specifier.

Inside protected access specifier.

Inside default access specifier.

Inside public access specifier.

Inside protected access specifier.

D:\Desktop\Java>java parent

Inside default access specifier.

Inside public access specifier.

Inside private access specifier.

Inside protected access specifier.

TABLE:

|  |  |  |  |
| --- | --- | --- | --- |
| Access Specifier | Same Class | Outside the Class | Subclass |
| Private | ☑ | × | × |
| Protected | ☑ | ☑ | ☑ |
| Public | ☑ | ☑ | ☑ |
| Default | ☑ | ☑ | ☑ |

1. Write a program in java to create a Number class that will accept a decimal number from user. Create a Binary and an Octal class that will inherit the Number class and convert the input number into binary and octal representations by using the respective method of the Binary and Octal classes.

class Number {

    int num;

    Number() {

        num = 0;

    }

    Number(int n) {

        num = n;

    }

    void convert() {

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

    }

}

class Binary extends Number {

    Binary() {

        super();

    }

    Binary(int n) {

        super(n);

    }

    void convert() {

        int temp = num;

        String ans = "";

        while(temp != 0) {

            ans = (temp % 2) + ans;

            temp /= 2;

        }

        System.out.println("Binary: " + ans);

    }

}

class Octal extends Number {

    Octal() {

        super();

    }

    Octal(int n) {

        super(n);

    }

    void convert() {

        int temp = num;

        String ans = "";

        while(temp != 0) {

            ans = (temp % 8) + ans;

            temp /= 8;

        }

        System.out.println("Octal: " + ans);

    }

}

class conversion {

    public static void main(String[] args) {

        int input = Integer.parseInt(args[0]);

        Number num = new Number(input);

        num.convert();

        Binary bin = new Binary(input);

        bin.convert();

        Octal oct = new Octal(input);

        oct.convert();

    }

}

OUTPUT:

Microsoft Windows [Version 10.0.19045.3324]

(c) Microsoft Corporation. All rights reserved.

C:\Users\Anurag Singh>d:

D:\>cd desktop

D:\Desktop>cd java

D:\Desktop\Java>javac fifteen.java

D:\Desktop\Java>java conversion 50

Decimal: 50

Binary: 110010

Octal: 62