**PRACTICAL – 3(7)**

**Aim:** Complete the code and write main () method to execute program.

        Public class MethodOverloading

    {

               Private void methodoverloading()

               {

                                //No argument,private method

                }

               Private int methodOverloaded(int i)

               {                            //code

}

               Protected  int methOdoverloaded(double d)

                {

**//code**

}

Public void methodOverloading(int i, double d)

                 {

                                              //code

 }

}

**PROGRAM CODE:**

public class Practical\_3\_7 {

    private *void* methodoverloaded() {

        System.out.println("private void mathodoverloaded():");

    }

    private *int* methodoverloaded(*int* *i*) {

        System.out.println("private int mathodoverloaded(int):");

        return 0;

    }

    private *void* methodoverloaded(*double* *i*) {

        System.out.println("private int mathodoverloaded(double):");

    }

    public *void* methodoverloaded(*int* *i*, *double* *j*) {

        System.out.println("public void mathodoverloaded(int,double):");

        System.out.println("\n20DCE019-Yatharth Chauhan");

    }

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

        Practical\_3\_7 s = new Practical\_3\_7();

        s.methodoverloaded();

        s.methodoverloaded(2);

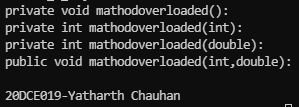
        s.methodoverloaded(12.22);

        s.methodoverloaded(2, 22.22);

    }

}

**OUTPUT:**

****

**CONCLUSION:** We get to know about the method overloading in Java.