Code:

//

//  main.cpp

//  MOSFET

//

//  Created by Abhinav Jayanthy on 4/17/15.

//  Copyright (c) 2015 Abhinav. All rights reserved.

//

#include <iostream>

#include <fstream>

#include <cmath>

#include <stdlib.h>

#include <graphics.h>

using namespace std;

int choice();

void help();

void drawMosfet();

void enhancementMode();

void depletionMode();

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*CLASS DECLARATION\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

class MOSFET{

private:

    double l;//Lenght of the transistor

    double w;// Width of the transistor

    double Vgs;//Gate source voltage

    double Tox;//Gate Oxide thickness

    double Cox;//Oxide Capacitance

    double m;//Mobility

    double Nb;//Channel doping

    double results[];//Results in array

public:

    //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*CONSTRUCTOR DECLARATION\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

    MOSFET();//Default Constructor

    MOSFET (double,double);//Construstor 1

    MOSFET(double, double, double ,double,double,double);// Constructor 2

    //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*ASSESSORS DECLARATION\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

    void set\_l (double x);

    double get\_l();

    void display\_l();

void set\_w(double x);

    double get\_w();

    void display\_w();

    void set\_Cox (double x);

    double get\_Cox();

    void display\_Cox();

    void set\_Tox (double x);

    double get\_Tox();

    void display\_Tox();

    void set\_Vgs(double x);

    double get\_Vgs();

    void display\_Vgs();

    void set\_m (double x);

    double get\_m();

    void display\_m();

    void set\_Nb(double x);

    double get\_Nb();

    void display\_Nb();

    //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*CUSTOM FUNCTION DECLARATION\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

    void display();

    void graphics();

    double transconducatance();

    double current();

    void set\_results(double array[]);

    // Destructor

     ~MOSFET(){}

};

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* END OF CLASS DECLARATION\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*CONSTRICTORS DEFENITIOS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

// Default Constructor

MOSFET :: MOSFET(){

    l = 1;

    w = 10;

    Tox = 20;

    Cox = (3.9\*8.85\*pow(10,-7))/Tox;

    m =300;

    Vgs=3;

    Nb = 2.129e15;

}

// Two parameters Constructor

MOSFET :: MOSFET(double length,double width){

    l = length;

    w = width;

    Tox = 20;

    Cox = (3.9\*8.85\*pow(10,-7))/Tox;

    m =300;

    Vgs=3;

    Nb = 2.129e15;

}

// All Parameters Constructor

MOSFET :: MOSFET(double length, double width, double OxideThickness,double mobility,double doping,double gateVoltage){

    l = length;

    w = width;

    Vgs = gateVoltage;

    Tox = OxideThickness;

    m = mobility;

    Nb = doping;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*END OF CONSTRUCTOR DEFENITIONS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*ASSESSORS DECLARATION\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

void MOSFET ::set\_l(double x){

    l = x;

}

double MOSFET :: get\_l(){

    return l;

}

void MOSFET ::set\_w(double x){

    w = x;

}

double MOSFET :: get\_w(){

    return w;

}

void MOSFET ::set\_Cox(double x){

    Cox = x;

}

double MOSFET :: get\_Cox(){

    return Cox;

}

void MOSFET ::set\_Tox(double x){

    Tox = x;

}

double MOSFET :: get\_Tox(){

    return Tox;

}

void MOSFET ::set\_m(double x){

    m = x;

}

double MOSFET :: get\_m(){

    return m;

}

void MOSFET ::set\_Nb(double x){

    Nb = x;

}

double MOSFET :: get\_Nb(){

    return Nb;

}

double MOSFET :: get\_Vgs(){

    return Vgs;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*CUSTOM FUNCTIONS BODY\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

// HELP FUNCTION

void help(){

    char answer;

    cout << "\n\nARE YOU NEW TO ELECTRONICS??\nIF YES WE TEACH WHAT A MOSFET IS ...[y/n]? ";

    cin >> answer;

    if (answer == 'y'||answer == 'Y') {

        string line;

        ifstream myfile("Help.txt");

        if ([myfile.is](http://myfile.is/)\_open()) {

            while (getline (myfile,line)) {

                cout << line<<endl;

            }

            cout <<"\n\n\t\t\t\t\t\tWELCOME\n";

        }

        else{

            cout << "FILE NOT FOUND"<<endl<<endl;

        }

    }

    else if (answer == 'n'|| answer == 'N'){

        cout <<"\n\n\t\t\t\t\t\tWELCOME\n";

    }

    else{

        cout <<"\n\nIt is either (Y)es(y) or (N)o(n)\n Please enter a valid option";

        help();

    }

}

// CHOICE FUNCTION

int choice(){

    int i;

    cout <<"\nHI, THIS PROGRAM CALCULATES THE TRANSCONDUCTANCE AND CURRENT FOR A MOSFET";

    cout << "\n\n\t\t\t\t\t\tHOW DO YOU WANT TO DESIGN YOUR MOSFET??"<<endl;

    cout << "1 --> CUSTOM DESIGN \n2 --> SEMI DESIGN \n3 --> EXAMPLE DESIGN\nANY OTHER TO EXIT THE PROGRAM"<<endl;

    cout << "\n\nENTER YOUR CHOICE  ";

    cin >>i;

    cout<<endl;

    return i;

}

//GRAPHICS

void MOSFET:: graphics(){

    char answer;

    int i;

    cout << "\nDo you wish to display any graphics?(y or n):";

    cin >> answer;

    if (answer == 'y' or answer == 'Y') {

        cout <<"\nDo you want the MOSFET in \n1-->Enhancemnt Mode\n2-->Depletion Mode"<<endl;

        cin>> i;

        if (i == 1) {

            enhancementMode();

            while (!kbhit()){

delay(2000);

    }

    closegraph();

    system ("PAUSE");

        }

        else if (i == 2){

            depletionMode();

            while (!kbhit()){

delay(2000);

    }

    closegraph();

    system ("PAUSE");

        }

        else{

            cout<<"Invalid option\nEnter again";

            graphics();

        }

    }

}

//Display function

void MOSFET::display(){

    cout<<"\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

    cout<< "\nThe Parameters of the transistor are :\n";

    cout<<"Channel Length = "<<get\_l()<<"\nChannel Width = "<<get\_w()<<"\nTox = "<<get\_Tox()<<"\nCox = "<<get\_Cox()<<"\nMobility = "<<get\_m()<<"\nVgs = "<<get\_Vgs()<<"\nChannel Doping = "<<get\_Nb();

    cout<<"\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

}

// THRESHOLD CALCULATION

double MOSFET:: transconducatance(){

    return m\*Cox\*(w/l)\*(Vgs - 0.7)\*1000;

}

// CURRENT CALCULATION

double MOSFET:: current(){

    return (m\*Cox\*(w/l)\*(pow((Vgs-0.7),2)/2))\*1000;

}

//Writing into a file using arrays

void MOSFET::set\_results(double x[]){

    ofstream myfile2("Results.txt",ios::app);

    myfile2<<"\*\*\*\*\*\*\*\*\*"<<endl<<"Input values"<<endl<<"Length = "<<get\_l()<<endl<<"Width = "<<get\_w()<<endl<<"Tox = "<<get\_Tox()<<endl<<"Cox = "<<get\_Cox()<<endl<<"Vgs = "<<get\_Vgs()<<endl<<"Mobility = "<<get\_m()<<endl<<"Doping Concentration = "<<get\_Nb()<<endl<<"\*\*\*\*\*\*\*\*"<<endl<<"Outputs"<<endl;

    for(int i = 0;i < 2 ;i++){

        if (i == 0) {

            myfile2 << "Transcondutance = "<<x[0]<<endl;

        }

        else if(i == 1){

            myfile2 << "Current = "<<x[1]<<endl;

        }

    }

}

//MOSFET TEMPLATE

void drawMosfet(){

    setfillstyle(1,GREEN);

    bar(800,250,450,350);

    setfillstyle(1,BLUE);

    bar(550,250,700,235);

    setfillstyle(1,YELLOW);

    bar(550,190,700,235);

    moveto(625,190);

    lineto(625,100);

    setcolor(12);

    circle(625,100,10);

    setfillstyle(1,RED);

    bar(450,250,570,280);

    moveto(500,250);

    setcolor(WHITE);

    lineto(500,150);

    setcolor(12);

    circle(500,150,10);

    setfillstyle(1,RED);

    bar(800,250,680,280);

    moveto(750,250);

    setcolor(WHITE);

    lineto(750,150);

    setcolor(12);

    circle(750 ,150,10);

}

void enhancementMode(){

initwindow(1500,700);

for(int y = 250;y<260;y=y+1){

cleardevice();

drawMosfet();

moveto(680,y);

setcolor(15);

lineto(568,y);

moveto(625,100);

// lineto(350,100);

delay(1000);

}

}

void depletionMode(){

initwindow(1500,700);

for(int y = 260;y>250;y=y-1){

cleardevice();

drawMosfet();

moveto(680,y);

setcolor(15);

lineto(568,y);

moveto(625,100);

//lineto(350,100);

delay(1000);

}

}

//Main function

int main() {

    int i;

    double temp[2];

    help();

    i=choice();

    while (i == 1|| i == 2 || i == 3) {

        if (i == 1) {

            double l,w,Tox,m,Nb,Vgs;//Lenght of the transistor

            // Call all parameter construstor

            //cout << "One selected"<<endl;

            MOSFET transistor;

            cout<< "\n\nEnter the Channel length of the Transistor(um) : ";

            cin>> l;

            transistor.set\_l(l);

            cout << "\n\nEnter the Channel width of the Transistor(um) : ";

            cin >> w;

            transistor.set\_w(w);

            cout << "\n\nEnter the Gate Oxide Thickness of the Transistor(nm) : ";

            cin >> Tox;

            transistor.set\_Tox(Tox);

            cout << "\n\nEnter the Mobility( cm2/V\*s) : ";

            cin >> m;

            transistor.set\_m(m);

            cout << "\n\nEnter the Channel Doping(1/cm3) : ";

            cin >> Nb;

            transistor.set\_Nb(Nb);

            cout << "\n\nEnter the Gate to source Voltage(V) : ";

            cin >> Vgs;

            transistor.set\_m(Vgs);

            cout<<"\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

            cout<<"\n\t\t\t\tRESULTS\n";

            cout <<"The Transcondutance(gm) is : "<< transistor.transconducatance()<<" mS"<<endl;

            cout <<"The Drain Current(Id) is : "<<transistor.current()<<" mA"<<endl;

            cout <<"\nCox = "<<transistor.get\_Cox()<<endl;

            cout<<"\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

            temp[0]=transistor.transconducatance();

            temp[1]=transistor.current();

            transistor.set\_results(temp);

            transistor.graphics();

            i=choice();

        }

        else if (i == 2){

            double l,w;

            MOSFET transistor(l,w);

            cout<< "\n\nEnter the Channel length of the Transistor(um) : ";

            cin>> l;

            transistor.set\_l(l);

            cout << "\n\nEnter the Channel width of the Transistor(um) : ";

            cin >> w;

            transistor.set\_w(w);

            cout<<"\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

            cout<<"\n\t\t\t\tRESULTS\n";

            cout <<"The Transcondutance(gm) is : "<< transistor.transconducatance()<<" mS"<<endl;

            cout <<"The Drain current(Id) is : "<<transistor.current()<<" mA"<<endl;

            cout <<"\nCox = "<<transistor.get\_Cox()<<endl;

            cout<<"\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

            temp[0]=transistor.transconducatance();

            temp[1]=transistor.current();

            transistor.set\_results(temp);

            transistor.graphics();

            i=choice();

        }

        else if (i == 3){

            MOSFET transistor;

            cout<<"\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

            cout<<"\n\t\t\t\tRESULTS\n";

            cout <<"The Transcondutance(gm) is : "<< transistor.transconducatance()<<" mS"<<endl;

            cout <<"The Drain Current(Id) is : "<<transistor.current()<<" mA"<<endl;

            cout<<"\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

            temp[0]=transistor.transconducatance();

            temp[1]=transistor.current();

            transistor.set\_results(temp);

            transistor.display();

            transistor.graphics();

            i=choice();

        }

    }

    cout << "\n\n\t\t\t\t\t\tSEE YOU !!\nPROJECT MOSFET \n\t\t\t\t\t-ABHINAV JAYANTHY (MIM289)";

    return 0;

}