

PRÁCTICA 5

Ejercicio 1:

```
#include <iostream>
#include <cstdlib>
#include <stdio.h>
#include <cstring>
#include <cmath>

using namespace std;

class CIRCULO //Clase Base
{
    protected:
        float Radio;
    public:
        CIRCULO();
        CIRCULO(float r);
        float Calcula_Area();
        void Imprime_Circulo();
};

//Clases Derivadas CILINDRO, ESFERA, CONO

//CILINDRO
class CILINDRO:public CIRCULO //Establece la herencia simple
{
    private:
        float Altura;
    public:
        CILINDRO();
        CILINDRO(float r, float h);
        float Calcula_Volumen();
        void Imprime_Cilindro();
};

//ESFERA
class ESFERA:public CIRCULO
{
    public:
        ESFERA();
        ESFERA(float r);
        float Calcula_VolumenE();
        void Imprime_Esfera();
};
```

```
};
```

```
//CONO
```

```
class CONO:public CIRCULO
{
    private:
        float AlturaC;
    public:
        CONO();
        CONO(float r, float hc);
        float Calcula_VolumenC();
        void Imprime_Conos();
};
```

```
//Definicion De Metodos Base CIRCULO
```

```
CIRCULO::CIRCULO()
```

```
{
    Radio=0;
}
```

```
CIRCULO::CIRCULO(float r)
```

```
{
    Radio=r;
}
```

```
float CIRCULO::Calcula_Area()
```

```
{
    return (M_PI*pow(Radio,2));
}
```

```
void CIRCULO::Imprime_Circulo()
```

```
{
    cout<<" \n  --CIRCULO--"<<endl;
    cout<<"El Area Del Circulo Es: "<<Calcula_Area()<<endl;
    cout<<"El Radio Del Circulo Es: "<<Radio<<endl;
}
```

```
//DEFINIENDO METODOS DE LA CLASE DERIVADA CILINDRO
```

```
CILINDRO::CILINDRO()
```

```
{
    Altura=0;
}
```

```

CILINDRO::CILINDRO(float r, float h):CIRCULO(r)
{
    Altura=h;
}

float CILINDRO::Calcula_Volumen()
{

    return (Calcula_Area()*Altura);

}

void CILINDRO::Imprime_Cilindro()
{
    cout<<"  \n --CILINDRO--"<<endl;
    cout<<"El Area Del Cilindro Es: "<<Calcula_Area()<<endl;
    cout<<"El Radio Del Cilindro Es: "<<Radio<<endl;
    cout<<"El Volumen Del Cilindro Es: "<<Calcula_Volumen()<<endl;
}

```

//Definiendo Los Metodos Derivados ESFERA

```

ESFERA::ESFERA()
{}

ESFERA::ESFERA(float r):CIRCULO (r)
{}

float ESFERA::Calcula_VolumenE()
{
    return ((M_PI*pow(Radio, 3)*4)/3);
}

void ESFERA::Imprime_Esfera()
{
    cout<<"  \n --ESFERA--"<<endl;
    cout<<"El Area Del Circulo Medio De La Esfera Es: "<<Calcula_Area()<<endl;
    cout<<"El Radio De La Esfera Es: "<<Radio<<endl;
    cout<<"El Volumen De La Esfera Es: "<<Calcula_VolumenE()<<endl;
}

```

```
//Definiendo Metodos Derivados CONO
```

```
CONO::CONO()  
{}
```

```
CONO::CONO(float r, float hc):CIRCULO(r)  
{  
    AlturaC=hc;  
}
```

```
float CONO::Calcula_VolumenC()  
{  
    return ((Calcula_Area()*AlturaC)/3);  
}
```

```
void CONO::Imprime_Conos()  
{  
    cout<<"  \n --CONO--"<<endl;  
    cout<<"El Area De La Base Del Cono Es: "<<Calcula_Area()<<endl;  
    cout<<"El Radio De La Base Del Cono Es: "<<Radio<<endl;  
    cout<<"El Volumen Del Cono Es: "<<Calcula_VolumenC()<<endl;  
}
```

```
//IMPLEMENTANDO LOS METODOS DE LAS CLASES
```

```
int main ()  
{  
    CIRCULO Circ1(2);  
    Circ1.Imprime_Circulo();  
    CILINDRO Cil1(2,5);  
    Cil1.Imprime_Cilindro();  
    ESFERA Esf1(2);  
    Esf1.Imprime_Esfera();  
    CONO Cono1(2,3);  
    Cono1.Imprime_Conos();  
  
    system("pause>null");  
}
```

Ejercicio 2:

```
#include <iostream>
#include <cstdlib>
#include <cmath>
using namespace std;

class INTEGRACION
{
protected:
    float a, b, vv, fa, fb, x2,x,in;
public:
    INTEGRACION();
    //void fx();
    void imprime();
};

INTEGRACION::INTEGRACION ()
{
    x2=3;//coeficiente de x^2
    x=6;//coeficiente de x
    in=3;//termino independiente
    vv=9;
    a=2;
    b=6;
    cout<<"Ingrese el limite inferior: ";
    cin>>a;
    cout<<"Ingrese el limite superior: ";
    cin>>b;
    cout<<endl;
    fa= (x2*(pow(a,2)))-(x*a)+in;
    fb= (x2*(pow(b,2)))-(x*b)+in;
}

void INTEGRACION:: imprime()
{
    cout<<endl<<"\t"<<"La Funcion Es: "<<x2<<"x^2- "<<x<<"x+ "<<in<<endl
    <<"\t"<<"El Limite Inferior Es: "<<a<<endl
    <<"\t"<<"El Limite Superior Es: "<<b<<endl
    <<"\t"<<"El Valor Verdadero De La Integral Es: "<<vv<<endl<<endl;
}

class trapecio:public INTEGRACION
{
private:
    float ls, lm;
```

```

    public:
        trapecio();

                                void ts();
                                void tm();

};

trapecio::trapecio()
{ ls=0; lm=0;}

void trapecio::ts()
{
    ls= (b-a)*(fa+fb)/2;
    imprime();
    cout<<"\t"<<"El Area Por Metodo De TRAPECIO SIMPLE Es: "<<ls<<endl<<endl
    <<"\t"<<"Error Verdadero: "<<ls-vv<<endl;
}

void trapecio::tm()
{
    float h, sum=0, fh;
    int n;
    cout<<"\t"<<"Ingrese El Numero De Segmentos (n): ";
    cin>>n;
    h=a+((b-a)/float (n));
    float h1=(b-a)/float (n);
    for(int i=1;i<n;i++)
    {
        fh=(x2*(pow(h,2)))-(x*h)+in;
        sum+=fh;
        h+=h1;
    }
    lm=(b-a)*((fa+(2*sum)+fb)/(2.0*n));
    imprime();
    cout<<"\t"<<"El Area Por Metodo De TRAPECIO MULTIPLE Es: "<<lm<<endl<<endl
    <<"\t"<<"Error Verdadero: "<<vv-lm<<endl;
}

class simpson:public INTEGRACION
{
    private:
        float ls1, ls3;

    public:
        simpson();
        void untercio();
        void tresoct();

```

```
};
```

```
simpson::simpson()  
{ ls1=0; ls3=0;  
}
```

```
void simpson::untercio()  
{  
    float xi=(b+a)/2.0;  
    float fxi=((x2*(pow(xi,2)))-(x*xi)+in)*4;  
    ls1=(b-a)*(fa+fxi+fb)/6.0;  
    imprime();  
    cout<<"\t"<<"El Area Por EL Metodo De SIMPSON 1/3 Es:"<<ls1<<endl<<endl  
        <<"\t"<<"Error Verdadero: " <<vv-ls1<<endl;  
}
```

```
void simpson::tresoct()  
{  
    float h=(b-a)/3, xj=h*2, fh, fxj;  
    fh=((x2*(pow(h,2)))-(x*h)+in)*3;  
    fxj=((x2*(pow(xj,2)))-(x*xj)+in)*3;  
    ls3=(b-a)*(fa+fxj+fh+fb)/8.0;  
    imprime();  
    cout<<"\t"<<"El Area Por Metodo De SIMPSON 3/8 ES:"<<ls3<<endl<<endl  
        <<"\t"<<"Error Verdadero: " <<vv-ls3<<endl;  
}
```

```
void menu()  
{  
    cout<<"\t"<<"\t"<<" --MENU-- " <<endl  
        <<"\t"<<"1. Trapecio Simple" <<endl  
        <<"\t"<<"2. Trapecio Multiple" <<endl  
        <<"\t"<<"3. Simpson 1/3" <<endl  
        <<"\t"<<"4. Simpson 3/8" <<endl;  
}
```

```
int main ()  
{  
    int op;  
    cout<<"\t"<<"LIMITES DE INTEGRACION POR --TRAPECIO-- : " <<endl;  
    trapecio obj1;  
    cout<<"\t"<<"LIMITES DE INTEGRACION POR --SIMPSON-- : " <<endl;  
    simpson obj2;  
    while (1)  
    {
```

```

menu();
cout<<endl<<"Elija El Metodo Por El Que Desea Aproximar: "<<"\t";
cin>>op;
switch (op)
{
    case 1:
        obj1.ts();
        system ("pause");
        system ("cls");
        break;
    case 2:
        obj1.tm();
        system ("pause");
        system ("cls");
        break;
    case 3:
        obj2.untercio();
        system ("pause");
        system ("cls");
        break;
    case 4:
        obj2.tresoct();
        system ("pause");
        system ("cls");
        break;
    default:
        cout<<"No Corresponde A Ningun Opcion. Prube De Nuevo"<<endl;
        system ("pause");
        system ("cls");
}
}
}

```