

Clases y Métodos

1. Ejecute el programa **devcpp.exe**.
2. En el menú **Archivo** seleccione la opción **Nuevo** y a continuación **Archivo Fuente**.
3. En el nuevo documento transcriba el código del siguiente programa, en el cual se define la clase **Potencia** con sus atributos (base y exponente) y sus métodos (constructor y de cálculo):

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

class Potencia {
    private: float base;
            int exponente;

    public: Potencia(float Dato1, int Dato2);
           Calcular_Potencia();
};

Potencia::Potencia(float Dato1, int Dato2) {
    base=Dato1;
    exponente=Dato2;
}

Potencia::Calcular_Potencia() {
    cout<<base<<"^"<<exponente<<"="<<pow(base,exponente)<<endl;
}

int main(void) {
    int expo;
    float base;
    cout<<endl<<endl<<"Ejemplifica el uso de clases sin plantillas"<<endl<<endl;
    cout<<"proporciona una base flotante: ";
    cin>>base;
    cout<<"proporciona un exponente entero: ";
    cin>>expo;
    Potencia Obj(base,expo);
    Obj.Calcular_Potencia();
}
```

4. Ahora localice el icono de **Compilar** que está en la barra de herramientas y oprímalo, asígnele el nombre de **ejercicio 2A** y elija el destino que guste para guardarlo.

5. Ahora localice el siguiente icono en la barra de herramientas que corresponde a **Ejecutar** y oprímalo.
6. Pruebe con diferentes valores para la base y el exponente, incluso si proporciona un número con decimales para el exponente, la aplicación conserva la parte entera para realizar el cálculo.
7. Ahora vamos a crear una nueva versión de este programa para permitirle utilizar otros tipos de datos sin tener que definir otra clase mediante plantillas:
8. En el menú **Archivo** seleccione la opción **Nuevo** y a continuación **Archivo Fuente**.
9. Copie completamente el código del ejemplo anterior en esta nueva ventana y edítelo para que coincida con el siguiente código (los cambios aparecen resaltados):

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

template<class T1, class T2>
class Potencia {
    private: T1 base;
            T2 exponente;

    public: Potencia(T1, T2);
           Calcular_Potencia();
};

template<class T1, class T2>
Potencia<T1,T2>::Potencia(T1 Dato1, T2 Dato2) {
    base=Dato1;
    exponente=Dato2;
}

template<class T1, class T2>
Potencia<T1,T2>::Calcular_Potencia() {
    cout<<base<<"^"<<exponente<<"="<<pow(base,exponente)<<endl;
}

int main(void) {
    double expo,base;
    cout<<endl<<endl<<"Ejemplifica el uso de plantillas de clases"<<endl<<endl;
    cout<<"proporciona una base doble: ";
    cin>>base;
    cout<<"proporciona un exponente doble: ";
    cin>>expo;
    Potencia<double, double> Obj(base,expo);
    Obj.Calcular_Potencia();
}
```

10. Ahora localice el icono de **Compilar** que está en la barra de herramientas y oprímalo, asígnele el nombre de **ejercicio 2B** y elija el destino que guste para guardarlo.
11. Ahora localice el siguiente icono en la barra de herramientas que corresponde a **Ejecutar** y oprímalo.
12. El funcionamiento es idéntico al programa anterior, pero en el programa principal se crea un objeto de la clase Potencia para números tipo **double**.
13. Ahora en el menú **Archivo** seleccione la nuevamente opción **Nuevo** y a continuación **Archivo Fuente**.
14. En el nuevo documento transcriba el código del siguiente programa, en el cual se define la clase **Círculo** con sus atributos (base y exponente) y sus métodos (constructor y de cálculo):

```
#include <iostream>
using namespace std;
class Circulo {
    public: Circulo ();
           void iniciar(void);
           void entradaDatos(void);
           void calcularArea(void);
           void calcularPerim(void);
           void salidaDatos(void);
           ~ Circulo ();
    private: double radio, area, perimetro, pi;
};

//Constructor de la clase Circulo
Circulo::Circulo () {
    pi=3.1416;
}

//Método que indica el inicio del programa
void Circulo::iniciar(void) {
    cout <<endl<<endl<<"Programa que ejemplifica la clase Circulo."<<endl<<endl;
}

//Método que permite la entrada de los datos
void Circulo::entradaDatos(void) {
    cout<<"Introduzca el radio: ";
    cin>>radio;
}

//Método que calcula el área de la figura círculo
void Circulo::calcularArea(void) {
    area = pi * radio * radio;
}
```

```
//Método que calcula el perímetro de la figura círculo
void Circulo::calcularPerim(void) {
    perimetro = 2 * pi * radio;
}

//Método que realiza la salida de los datos
void Circulo::salidaDatos(void) {
    cout<< "Area: " <<area<<endl;
    cout<< "Perimetro: " <<perimetro<<endl;
}

//Destructor de la clase Circulo
Circulo::~Circulo () {}

int main(void) {
    Circulo obj;
    obj.iniciar();
    obj.entradaDatos();
    obj.calcularArea();
    obj.calcularPerim();
    obj.salidaDatos();
}
```

15. Ahora localice el icono de **Compilar** que está al inicio de la barra de herramientas y oprímalo, proporcione el nombre de **ejercicio 2C** y elija el destino que guste para guardarlo.
16. Ahora ejecútelo varias veces con distintos valores del radio, observe que incluso funciona con valores negativos.
17. Las estructuras son tipos de datos abstractos, que a su vez son clases, por lo que podemos usar indistintamente la palabra **struct** en vez de **class**.
18. Por lo tanto, en la declaración de la clase **Círculo** reemplace la palabra **class** por **struct** y vuelva a compilar el programa.
19. Observe que la ejecución del programa es idéntica.
20. Fin de la Práctica.