

Laboratorio 08

Polimorfismo

Nombre: Gabriel Fernando Rodriguez Cutimbo

CUI: 20212157

Grupo: B

Repositorio GitHub:

https://github.com/gaco123/EPCC_CCII.git

1. Ejercicios

Resolver los siguientes ejercicios planteados:

1. Defina una clase Forma que tenga los siguientes miembros de datos:

- Color
- Coordenada del centro de la forma (objeto Punto)
- Nombre de la forma (char *)

Y, al menos, las siguientes funciones miembro:

- Imprimir
- Obtener y cambiar el color
- Mover la forma (o sea, su centro)

Defina una clase derivada Rectángulo que tenga los siguientes miembros como datos:

- Lado menor.
- Lado mayor.

Y, al menos, las siguientes funciones miembro:

- Imprimir. Debe imprimir qué se trata de un rectángulo mostrando su nombre, color, centro y lado. Debería usarse la función Imprimir de la clase base para realizar parte de este trabajo.
- Calcular el área (lado menor * lado mayor).
- Calcular el perímetro. ($2 * \text{lado menor} + 2 * \text{lado mayor}$).
- Cambiar el tamaño del rectángulo. Recibe como parámetro un factor de escala. Así, por ejemplo, si el factor vale 2, el rectángulo duplicará su tamaño y si es 0,5 se reducirá a la mitad.

Realice un programa que pruebe el funcionamiento de estas clases. Debe crear objetos y comprobar el correcto funcionamiento de las funciones miembro.

Código “Punto.h”:

```
#ifndef PUNTO_H
#define PUNTO_H
#include <iostream>
using namespace std;

class Punto{
public:
    Punto();
    Punto(Punto const&);
    Punto(double,double);
    ~Punto();
    void camb_punto(double,double);
    void imprimir();
protected:
    double x;
    double y;
};

#endif
```

Código “Punto.cpp”:

```
#include "Punto.h"

Punto::Punto()=default;
Punto::Punto(const Punto &x)=default;
Punto::Punto(double _x, double _y){
    x=_x;
    y=_y;
}
Punto::~~Punto(){
}
void Punto::camb_punto(double _x, double _y){
    x=_x;
    y=_y;
    cout<<"La coordenada X ha cambiado a: "<< x <<endl;
    cout<<"La coordenada Y ha cambiado a: "<< y <<endl;
}
void Punto::imprimir(){
    cout<<"La coordenada X es: "<< x <<endl;
    cout<<"La coordenada Y es: "<< y <<endl;
}
```

Código “Forma.h”:

```
#ifndef FORMA_H
#define FORMA_H
```

```
#define _USE_MATH_DEFINES
#include <string>
#include <cmath>
#include "Punto.h"

class Forma: public Punto{
public:
    Forma();
    Forma(Forma const&);
    Forma(string,double,double,char[]);
    ~Forma();
    void imprimir();
    string obt_color();
    void camb_color(string);
protected:
    string color;
    char* nombre;
};

#endif
```

Código “Forma.cpp”:

```
#include "Forma.h"

Forma::Forma()=default;
Forma::Forma(const Forma &x)=default;
Forma::Forma(string _color, double _x, double _y, char _nombre[])
:Punto(_x,_y){
    color=_color;
    nombre=_nombre;
}
Forma::~Forma(){
    delete[] nombre;
}
void Forma::imprimir(){
    cout<<"El nombre es: "<< nombre <<endl;
    cout<<"El color es: "<< color <<endl;
    Punto::imprimir();
}
string Forma::obt_color(){
    return color;
}
void Forma::camb_color(string _color){
    color=_color;
}
```

Código “Rectangulo.h”:

```
#ifndef RECTANGULO_H
#define RECTANGULO_H
#include "Forma.h"
```

```
class Rectangulo : public Forma{
public:
    Rectangulo();
    Rectangulo(Rectangulo const&);
    Rectangulo(string, double, double, char[], double, double);
    ~Rectangulo();
    void imprimir();
    double Area();
    double Perimetro();
    void Escalar(double);
protected:
    double lmenor;
    double lmayor;
};

#endif
```

Código “Rectangulo.cpp”:

```
#include "Rectangulo.h"

Rectangulo::Rectangulo()=default;
Rectangulo::Rectangulo(const Rectangulo &x)=default;
Rectangulo::Rectangulo(string _color, double _x, double _y, char _nombre[], double _lmenor, double _lmayor)
:_Forma(_color,_x,_y,_nombre){
    lmenor=_lmenor;
    lmayor=_lmayor;
}
Rectangulo::~~Rectangulo(){
}
void Rectangulo::imprimir(){
    Forma::imprimir();
    cout<<"El lado menor es: "<< lmenor <<endl;
    cout<<"El lado mayor es: "<< lmayor <<endl;
}
double Rectangulo::Area(){
    return lmenor*lmayor;
}
double Rectangulo::Perimetro(){
    return (2*lmenor)+(2*lmayor);
}
void Rectangulo::Escalar(double es){
    lmenor=lmenor*es;
    lmayor=lmayor*es;
    cout<<"Los lados han sido escalados por "<<es<<endl;
}
```

Código “Lab_08.cpp”:

```
#include <iostream>
```

```
#include <string>
#include <locale>//Caracteres Español
#include "Punto.h"
#include "Forma.h"
#include "Rectangulo.h"
using namespace std;

int main (){
    setlocale(LC_CTYPE, "Spanish");//Caracteres Español

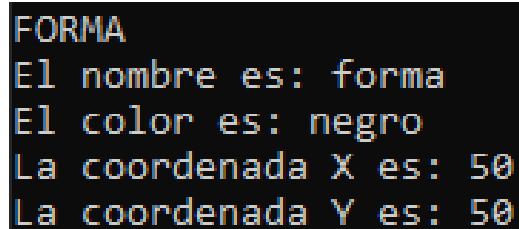
    //Forma
    char n1[]={ "forma"};
    Forma a("negro",50,50,n1);
    cout<<"FORMA\n";
    a.imprimir();
    cout<<"\n";

    //Rectangulo 1er-Parte
    char n2[]={ "rectangulo"};
    Rectangulo b("azul",1,0,n2,5,10);
    cout<<"RECTANGULO-1er\n";
    b.imprimir();
    cout<<"El área es: "<<b.Area()<<endl;
    cout<<"El perimetro es: "<<b.Perimetro()<<endl;
    cout<<"\n";

    //Rectangulo 2da-Parte
    cout<<"RECTANGULO-2do\n";
    b.camb_punto(0,1);
    b.Escalar(0.5);
    b.imprimir();
    cout<<"El área es: "<<b.Area()<<endl;
    cout<<"El perimetro es: "<<b.Perimetro()<<endl;
    cout<<"\n";

    return 0;
}
```

Funcionamiento (Forma):



```
FORMA
El nombre es: forma
El color es: negro
La coordenada X es: 50
La coordenada Y es: 50
```

Funcionamiento (Rectángulo):

```
RECTANGULO-1er
El nombre es: rectangulo
El color es: azul
La coordenada X es: 1
La coordenada Y es: 0
El lado menor es: 5
El lado mayor es: 10
El área es: 50
El perímetro es: 30

RECTANGULO-2do
La coordenada X ha cambiado a: 0
La coordenada Y ha cambiado a: 1
Los lados han sido escalados por 0.5
El nombre es: rectangulo
El color es: azul
La coordenada X es: 0
La coordenada Y es: 1
El lado menor es: 2.5
El lado mayor es: 5
El área es: 12.5
El perímetro es: 15
```

2. Defina una clase Elipse derivada de forma. Recordatorio: una elipse queda definida por su radio mayor (R) y su radio menor (r), tal que el área de una elipse es igual a $\pi \cdot (R \cdot r)$.

Código “Elipse.h”:

```
#ifndef ELIPSE_H
#define ELIPSE_H
#include "Forma.h"

class Elipse : public Forma{
public:
    Elipse();
    Elipse(Elipse const&);
    Elipse(string, double, double, char[], double, double);
    ~Elipse();
    void imprimir();
    double Area();
    void Escalar(double);
protected:
    double rmenor;
    double rmayor;
};

#endif
```

Código “Ellipse.cpp”:

```
#include "Ellipse.h"

Ellipse::Ellipse()=default;
Ellipse::Ellipse(const Ellipse &x)=default;
Ellipse::Ellipse(string _color, double _x, double _y, char _nombre[], double _rmenor, double _rmayor)
:Forma(_color,_x,_y,_nombre){
    rmenor=_rmenor;
    rmayor=_rmayor;
}
Ellipse::~~Ellipse(){
}
void Ellipse::imprimir(){
    Forma::imprimir();
    cout<<"El radio menor es: "<< rmenor <<endl;
    cout<<"El radio mayor es: "<< rmayor <<endl;
}
double Ellipse::Area(){
    return rmenor*rmayor*M_PI;
}
void Ellipse::Escalar(double es){
    rmenor=rmenor*es;
    rmayor=rmayor*es;
    cout<<"Los radios han sido escalados por "<<es<<endl;
}
```

Código “Lab_08.cpp”:

```
#include <iostream>
#include <string>
#include <locale>//Caracteres Español
#include "Punto.h"
#include "Forma.h"
#include "Ellipse.h"
using namespace std;

int main (){
    setlocale(LC_CTYPE, "Spanish");//Caracteres Español

    //Ellipse 1er-Parte
    char n3[]=("ellipse");
    Ellipse c("verde",-1,-1,n3,10,20);
    cout<<"ELIPSE-1er\n";
    c.imprimir();
    cout<<"El área es: "<<c.Area()<<endl;
    cout<<"\n";

    //Ellipse 2da-Parte
    cout<<"ELIPSE-2do\n";
    c.camb_punto(0,0);
    c.Escalar(2);
}
```

```
c.imprimir();  
cout<<"El área es: "<<c.Area()<<endl;  
cout<<"\n";  
  
return 0;  
}
```

Funcionamiento (Ellipse):

```
ELIPSE-1er  
El nombre es: ellipse  
El color es: verde  
La coordenada X es: -1  
La coordenada Y es: -1  
El radio menor es: 10  
El radio mayor es: 20  
El área es: 628.319  
  
ELIPSE-2do  
La coordenada X ha cambiado a: 0  
La coordenada Y ha cambiado a: 0  
Los radios han sido escalados por 2  
El nombre es: ellipse  
El color es: verde  
La coordenada X es: 0  
La coordenada Y es: 0  
El radio menor es: 20  
El radio mayor es: 40  
El área es: 2513.27
```

3. Defina una clase Cuadrado derivada de la clase Rectángulo.

Código “Cuadrado.h”:

```
#ifndef CUADRADO_H  
#define CUADRADO_H  
#include "Rectangulo.h"  
  
class Cuadrado : public Rectangulo{  
public:  
    Cuadrado();  
    Cuadrado(Cuadrado const&);  
    Cuadrado(string, double, double, char[], double);  
    ~Cuadrado();  
    void imprimir();  
};  
  
#endif
```

Código “Cuadrado.cpp”:

```
#include "Cuadrado.h"
```



```
Cuadrado::Cuadrado()=default;
Cuadrado::Cuadrado(const Cuadrado &x)=default;
Cuadrado::Cuadrado(string _color, double _x, double _y, char _nombre[], double _lado)
:Rectangulo(_color,_x,_y,_nombre,_lado,_lado){
}
Cuadrado::~~Cuadrado(){
}
void Cuadrado::imprimir(){
    Forma::imprimir();
    cout<<"El lado es: "<< lmenor <<endl;
}
```

Código “Lab_08.cpp”:

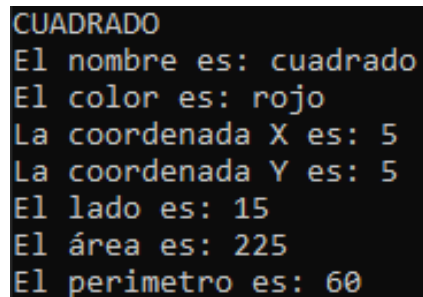
```
#include <iostream>
#include <string>
#include <locale> //Caracteres Español
#include "Punto.h"
#include "Forma.h"
#include "Rectangulo.h"
#include "Cuadrado.h"
using namespace std;

int main (){
    setlocale(LC_CTYPE, "Spanish");//Caracteres Español

    //Cuadrado
    char n4[]=("cuadrado");
    Cuadrado d("rojo",5,5,n4,15);
    cout<<"CUADRADO\n";
    d.imprimir();
    cout<<"El área es: "<<d.Area()<<endl;
    cout<<"El perimetro es: "<<d.Perimetro()<<endl;
    cout<<"\n";

    return 0;
}
```

Funcionamiento (Cuadrado):



```
CUADRADO
El nombre es: cuadrado
El color es: rojo
La coordenada X es: 5
La coordenada Y es: 5
El lado es: 15
El área es: 225
El perimetro es: 60
```

-
4. Defina una clase Circulo derivada de la clase Elipse.

Código “Circulo.h”:

```
#ifndef CIRCULO_H
#define CIRCULO_H
#include "Elipse.h"

class Circulo : public Elipse{
public:
    Circulo();
    Circulo(Circulo const&);
    Circulo(string, double, double, char[], double);
    ~Circulo();
    void imprimir();
};

#endif
```

Código “Circulo.cpp”:

```
#include "Circulo.h"

Circulo::Circulo()=default;
Circulo::Circulo(const Circulo &x)=default;
Circulo::Circulo(string _color, double _x, double _y, char _nombre[], double _radio)
:Elipse(_color,_x,_y,_nombre,_radio,_radio){
}
Circulo::~Circulo(){
}
void Circulo::imprimir(){
    Forma::imprimir();
    cout<<"El radio es: "<< rmenor <<endl;
}
```

Código “Lab_08.cpp”:

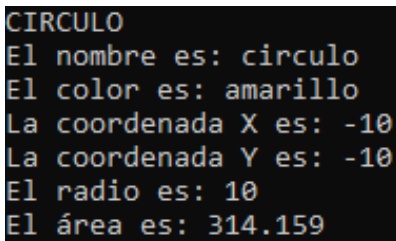
```
#include <iostream>
#include <string>
#include <locale> //Cáracteres Español
#include "Punto.h"
#include "Forma.h"
#include "Elipse.h"
#include "Circulo.h"
using namespace std;

int main (){
    setlocale(LC_CTYPE, "Spanish");//Cáracteres Español

    //Circulo
    char n5[]="circulo";
```

```
Circulo e("amarillo",-10,-10,n5,10);  
cout<<"CIRCULO\n";  
e.imprimir();  
cout<<"El área es: "<<e.Area()<<endl;  
cout<<"\n";  
  
return 0;  
}
```

Funcionamiento (Circulo):



```
CIRCULO  
El nombre es: circulo  
El color es: amarillo  
La coordenada X es: -10  
La coordenada Y es: -10  
El radio es: 10  
El área es: 314.159
```

5. Realice un programa que defina varias formas diferentes, cree un vector de punteros de la clase Forma que apunten a los objetos creados. El programa debe realizar un bucle que recorra todas las formas, las ponga todas del mismo color y las mueva a una determinada posición.

Código “Lab_08”:

```
#include <iostream>  
#include <string>  
#include <locale>//Caracteres Español  
#include "Punto.h"  
#include "Forma.h"  
#include "Rectangulo.h"  
#include "Elipse.h"  
#include "Cuadrado.h"  
#include "Circulo.h"  
using namespace std;  
  
int main (){  
    setlocale(LC_CTYPE, "Spanish");//Caracteres Español  
  
    //Rectangulo 1er-Parte  
    char n2[]={ "rectangulo"};  
    Rectangulo b("azul",1,0,n2,5,10);  
    cout<<"RECTANGULO-1er\n";  
    b.imprimir();  
    cout<<"El área es: "<<b.Area()<<endl;  
    cout<<"El perimetro es: "<<b.Perimetro()<<endl;  
    cout<<"\n";
```

```
//Rectangulo 2da-Parte
cout<<"RECTANGULO-2do\n";
b.camb_punto(0,1);
b.Escalar(0.5);
b.imprimir();
cout<<"El área es: "<<b.Area()<<endl;
cout<<"El perimetro es: "<<b.Perimetro()<<endl;
cout<<"\n";

//Elipse 1er-Parte
char n3[]=("elipse");
Elipse c("verde",-1,-1,n3,10,20);
cout<<"ELIPSE-1er\n";
c.imprimir();
cout<<"El área es: "<<c.Area()<<endl;
cout<<"\n";

//Elipse 2da-Parte
cout<<"ELIPSE-2do\n";
c.camb_punto(0,0);
c.Escalar(2);
c.imprimir();
cout<<"El área es: "<<c.Area()<<endl;
cout<<"\n";

//Cuadrado
char n4[]=("cuadrado");
Cuadrado d("rojo",5,5,n4,15);
cout<<"CUADRADO\n";
d.imprimir();
cout<<"El área es: "<<d.Area()<<endl;
cout<<"El perimetro es: "<<d.Perimetro()<<endl;
cout<<"\n";

//Circulo
char n5[]="circulo";
Circulo e("amarillo",-10,-10,n5,10);
cout<<"CIRCULO\n";
e.imprimir();
cout<<"El área es: "<<e.Area()<<endl;
cout<<"\n";

//Vector de Formas
int n = 4;
Forma** vec = new Forma*[n];
vec[0]=&b;//Rectangulo
vec[1]=&c;//Elipse
vec[2]=&d;//Cuadrado
vec[3]=&e;//Circulo

//Cambiar color a los objetos en el vector y moverlos
cout<<"CAMBIO DE COLOR Y POSICION PARA LAS 4 FORMAS\n";
for(int i=0; i<n; i++){
```

```
        vec[i]->camb_color("café");  
        cout<<"vec["<<i<<"]="<<vec[i]->obt_color()<<"\n";  
        vec[i]->camb_punto(100,100);  
    }  
    cout<<endl;  
  
    return 0;  
}
```

Funcionamiento (Rectángulo, Elipse, Cuadrado y Círculo):

```
RECTANGULO-1er  
El nombre es: rectangulo  
El color es: azul  
La coordenada X es: 1  
La coordenada Y es: 0  
El lado menor es: 5  
El lado mayor es: 10  
El área es: 50  
El perímetro es: 30  
  
RECTANGULO-2do  
La coordenada X ha cambiado a: 0  
La coordenada Y ha cambiado a: 1  
Los lados han sido escalados por 0.5  
El nombre es: rectangulo  
El color es: azul  
La coordenada X es: 0  
La coordenada Y es: 1  
El lado menor es: 2.5  
El lado mayor es: 5  
El área es: 12.5  
El perímetro es: 15
```

```
ELIPSE-1er  
El nombre es: elipse  
El color es: verde  
La coordenada X es: -1  
La coordenada Y es: -1  
El radio menor es: 10  
El radio mayor es: 20  
El área es: 628.319  
  
ELIPSE-2do  
La coordenada X ha cambiado a: 0  
La coordenada Y ha cambiado a: 0  
Los radios han sido escalados por 2  
El nombre es: elipse  
El color es: verde  
La coordenada X es: 0  
La coordenada Y es: 0  
El radio menor es: 20  
El radio mayor es: 40  
El área es: 2513.27
```

```
CUADRADO
El nombre es: cuadrado
El color es: rojo
La coordenada X es: 5
La coordenada Y es: 5
El lado es: 15
El área es: 225
El perimetro es: 60

CIRCULO
El nombre es: circulo
El color es: amarillo
La coordenada X es: -10
La coordenada Y es: -10
El radio es: 10
El área es: 314.159
```

Funcionamiento (Cambio de Color y Posición):

```
CAMBIO DE COLOR Y POSICION PARA LAS 4 FORMAS
vec[0]=café
La coordenada X ha cambiado a: 100
La coordenada Y ha cambiado a: 100
vec[1]=café
La coordenada X ha cambiado a: 100
La coordenada Y ha cambiado a: 100
vec[2]=café
La coordenada X ha cambiado a: 100
La coordenada Y ha cambiado a: 100
vec[3]=café
La coordenada X ha cambiado a: 100
La coordenada Y ha cambiado a: 100
```

6. Analice qué ocurre en el ejercicio anterior si se intenta imprimir la información de cada forma y qué sucede si se intenta obtener en ese bucle el área de todas las formas del vector.
7. Utilice la técnica de las funciones virtuales para arreglar los comportamientos anómalos detectados en el ejercicio anterior.

Código “Lab_08” (void imprimir()):

```
#include <iostream>
#include <string>
#include <locale> //Caracteres Español
```

```
#include "Punto.h"
#include "Forma.h"
#include "Rectangulo.h"
#include "Elipse.h"
#include "Cuadrado.h"
#include "Circulo.h"
using namespace std;

int main (){
    setlocale(LC_CTYPE, "Spanish");//Cáracteres Español

    //Rectangulo 1er-Parte
    char n2[]={ "rectangulo" };
    Rectangulo b("azul",1,0,n2,5,10);
    cout<<"RECTANGULO-1er\n";
    b.imprimir();
    cout<<"El área es: "<<b.Area()<<endl;
    cout<<"El perimetro es: "<<b.Perimetro()<<endl;
    cout<<"\n";

    //Rectangulo 2da-Parte
    cout<<"RECTANGULO-2do\n";
    b.camb_punto(0,1);
    b.Escalar(0.5);
    b.imprimir();
    cout<<"El área es: "<<b.Area()<<endl;
    cout<<"El perimetro es: "<<b.Perimetro()<<endl;
    cout<<"\n";

    //Elipse 1er-Parte
    char n3[]={ "elipse" };
    Elipse c("verde",-1,-1,n3,10,20);
    cout<<"ELIPSE-1er\n";
    c.imprimir();
    cout<<"El área es: "<<c.Area()<<endl;
    cout<<"\n";

    //Elipse 2da-Parte
    cout<<"ELIPSE-2do\n";
    c.camb_punto(0,0);
    c.Escalar(2);
    c.imprimir();
    cout<<"El área es: "<<c.Area()<<endl;
    cout<<"\n";

    //Cuadrado
    char n4[]={ "cuadrado" };
    Cuadrado d("rojo",5,5,n4,15);
    cout<<"CUADRADO\n";
    d.imprimir();
    cout<<"El área es: "<<d.Area()<<endl;
    cout<<"El perimetro es: "<<d.Perimetro()<<endl;
    cout<<"\n";
```

```
//Circulo
char n5[]="circulo";
Circulo e("amarillo",-10,-10,n5,10);
cout<<"CIRCULO\n";
e.imprimir();
cout<<"El área es: "<<e.Area()<<endl;
cout<<"\n";

//Vector de Formas
int n = 4;
Forma** vec = new Forma*[n];
vec[0]=&b;//Rectangulo
vec[1]=&c;//Elipse
vec[2]=&d;//Cuadrado
vec[3]=&e;//Circulo

//Cambiar color a los objetos en el vector y moverlos
cout<<"CAMBIO DE COLOR Y POSICION PARA LAS 4 FORMAS\n";
for(int i=0; i<n; i++){
    vec[i]->camb_color("café");
    cout<<"vec["<<i<<"]="<<vec[i]->obt_color()<<"\n";
    vec[i]->camb_punto(100,100);
}
cout<<endl;

//Verificar el imprimir()
for(int i=0; i<n; i++){
    cout<<"FORMA EN V["<<i<<"]\n";
    vec[i]->imprimir();
    cout<<endl;
}

return 0;
}
```

Funcionamiento (Imprimir desde el vector de punteros):

```
FORMA EN V[0]
El nombre es: rectangulo
El color es: café
La coordenada X es: 100
La coordenada Y es: 100

FORMA EN V[1]
El nombre es: elipse
El color es: café
La coordenada X es: 100
La coordenada Y es: 100

FORMA EN V[2]
El nombre es: cuadrado
El color es: café
La coordenada X es: 100
La coordenada Y es: 100

FORMA EN V[3]
El nombre es: circulo
El color es: café
La coordenada X es: 100
La coordenada Y es: 100
```

Análisis (Falla al imprimir):

Esto sucede debido a que en Forma la función void imprimir() no está declarada como virtual y por eso a la hora de imprimir cada objeto dentro del vector de punteros, no esta tomando la función de cada clase del objeto correspondiente sino que esta tomando la función void imprimir() de Forma, debido al orden que maneja el compilador a la hora de llamar una función que tiene el mismo nombre en las clase hijas y la clase base.

Solución (Falla al imprimir) “Forma.h”:

```
#ifndef FORMA_H
#define FORMA_H
#define _USE_MATH_DEFINES
#include <string>
#include <cmath>
#include "Punto.h"

class Forma: public Punto{
public:
    Forma();
    Forma(Forma const&);
    Forma(string,double,double,char[]);
    ~Forma();
    virtual void imprimir(); //la solución es poner esto en vez de solo “void imprimir();”
    string obt_color();
    void camb_color(string);
protected:
    string color;
    char* nombre;
};

#endif
```

Funcionamiento (Solución-parte-1-void imprimir()):

```
FORMA EN V[0]
El nombre es: rectangulo
El color es: café
La coordenada X es: 100
La coordenada Y es: 100
El lado menor es: 2.5
El lado mayor es: 5

FORMA EN V[1]
El nombre es: elipse
El color es: café
La coordenada X es: 100
La coordenada Y es: 100
El radio menor es: 20
El radio mayor es: 40
```

Funcionamiento (Solución-parte-2-void imprimir()):

```
FORMA EN V[2]
El nombre es: cuadrado
El color es: café
La coordenada X es: 100
La coordenada Y es: 100
El lado es: 15

FORMA EN V[3]
El nombre es: circulo
El color es: café
La coordenada X es: 100
La coordenada Y es: 100
El radio es: 10
```

Código “Lab_08” (double Area()):

```
#include <iostream>
#include <string>
#include <locale> //Caracteres Español
#include "Punto.h"
#include "Forma.h"
#include "Rectangulo.h"
#include "Elipse.h"
#include "Cuadrado.h"
#include "Circulo.h"
using namespace std;

int main (){
    setlocale(LC_CTYPE, "Spanish"); //Caracteres Español

    //Rectangulo 1er-Parte
    char n2[]={ "rectangulo" };
    Rectangulo b("azul",1,0,n2,5,10);
    cout<<"RECTANGULO-1er\n";
    b.imprimir();
    cout<<"El área es: "<<b.Area()<<endl;
    cout<<"El perimetro es: "<<b.Perimetro()<<endl;
    cout<<"\n";

    //Rectangulo 2da-Parte
    cout<<"RECTANGULO-2do\n";
    b.camb_punto(0,1);
    b.Escalar(0.5);
    b.imprimir();
    cout<<"El área es: "<<b.Area()<<endl;
    cout<<"El perimetro es: "<<b.Perimetro()<<endl;
    cout<<"\n";
```

```
//Elipse 1er-Parte
char n3[]=("elipse");
Elipse c("verde",-1,-1,n3,10,20);
cout<<"ELIPSE-1er\n";
c.imprimir();
cout<<"El área es: "<<c.Area()<<endl;
cout<<"\n";

//Elipse 2da-Parte
cout<<"ELIPSE-2do\n";
c.camb_punto(0,0);
c.Escalar(2);
c.imprimir();
cout<<"El área es: "<<c.Area()<<endl;
cout<<"\n";

//Cuadrado
char n4[]=("cuadrado");
Cuadrado d("rojo",5,5,n4,15);
cout<<"CUADRADO\n";
d.imprimir();
cout<<"El área es: "<<d.Area()<<endl;
cout<<"El perimetro es: "<<d.Perimetro()<<endl;
cout<<"\n";

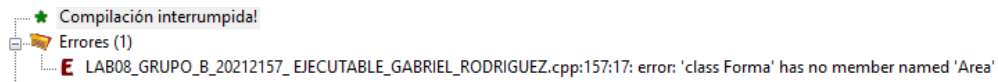
//Circulo
char n5[]=("circulo");
Circulo e("amarillo",-10,-10,n5,10);
cout<<"CIRCULO\n";
e.imprimir();
cout<<"El área es: "<<e.Area()<<endl;
cout<<"\n";

//Vector de Formas
int n = 4;
Forma** vec = new Forma*[n];
vec[0]=&b;//Rectangulo
vec[1]=&c;//Elipse
vec[2]=&d;//Cuadrado
vec[3]=&e;//Circulo

//Obtener áreas
cout<<"ÁREAS DE LOS OBJETOS\n";
double ab[]={0,0,0,0};
for(int i=0; i<n; i++){
    ab[i]=vec[i]->Area();
    if(i!=3){
        cout<<ab[i]<<" ";
    }
    else{
        cout<<ab[i]<<endl;
    }
}
```

```
    }  
  
    return 0;  
}
```

Funcionamiento (Error al intentar llamar la función double Area()):



Análisis (Falla al encontrar la función double Area()):

Debido a que esta función no está declarada en “Forma.h” a la hora de llamarla el compilador nos devolverá un error.

Solución (Falla al encontrar la función double Area()) “Forma.h”:

```
#ifndef FORMA_H  
#define FORMA_H  
#define _USE_MATH_DEFINES  
#include <string>  
#include <cmath>  
#include "Punto.h"  
  
class Forma: public Punto{  
public:  
    Forma();  
    Forma(Forma const&);  
    Forma(string,double,double,char[]);  
    ~Forma();  
    virtual void imprimir();  
    virtual double Area(); //se agrega esta línea de código para que el vector de punteros a Forma reconozca  
    la función “double Area();”  
    string obt_color();  
    void camb_color(string);  
protected:  
    string color;  
    char* nombre;  
};  
  
#endif
```

Solución (Falla al encontrar la función double Area()) “Forma.cpp”:

```
#include "Forma.h"  
  
Forma::Forma()=default;  
Forma::Forma(const Forma &x)=default;  
Forma::Forma(string _color, double _x, double _y, char _nombre[])  
:Punto(_x,_y){
```

```
        color=_color;
        nombre=_nombre;
    }
    Forma::~Forma(){
        delete[] nombre;
    }
    void Forma::imprimir(){
        cout<<"El nombre es: "<< nombre <<endl;
        cout<<"El color es: "<< color <<endl;
        Punto::imprimir();
    }
    double Forma::Area(){
        //Se agrega esto en el .cpp debido a que en el .h se ha declarado esta función
    }
    string Forma::obt_color(){
        return color;
    }
    void Forma::camb_color(string _color){
        color=_color;
    }
}
```

Funcionamiento (Solución double Area()):

```
ÁREAS DE LOS OBJETOS
12.5, 2513.27, 225, 314.159
```

8. Desarrolle un programa que, dado un conjunto de formas, calcule cuál tiene el área máxima e imprima la información de dicha forma.

Código “Lab_08.cpp”:

```
#include <iostream>
#include <string>
#include <locale> //Caracteres Español
#include "Punto.h"
#include "Forma.h"
#include "Rectangulo.h"
#include "Elipse.h"
#include "Cuadrado.h"
#include "Circulo.h"
using namespace std;

int main (){
    setlocale(LC_CTYPE, "Spanish");//Caracteres Español

    //Rectangulo 1er-Parte
    char n2[]={ "rectangulo" };
    Rectangulo b("azul",1,0,n2,5,10);
    cout<<"RECTANGULO-1er\n";
```

```
b.imprimir();
cout<<"El área es: "<<b.Area()<<endl;
cout<<"El perimetro es: "<<b.Perimetro()<<endl;
cout<<"\n";
```

```
//Rectangulo 2da-Parte
cout<<"RECTANGULO-2do\n";
b.camb_punto(0,1);
b.Escalar(0.5);
b.imprimir();
cout<<"El área es: "<<b.Area()<<endl;
cout<<"El perimetro es: "<<b.Perimetro()<<endl;
cout<<"\n";
```

```
//Elipse 1er-Parte
char n3[]=("elipse");
Elipse c("verde",-1,-1,n3,10,20);
cout<<"ELIPSE-1er\n";
c.imprimir();
cout<<"El área es: "<<c.Area()<<endl;
cout<<"\n";
```

```
//Elipse 2da-Parte
cout<<"ELIPSE-2do\n";
c.camb_punto(0,0);
c.Escalar(2);
c.imprimir();
cout<<"El área es: "<<c.Area()<<endl;
cout<<"\n";
```

```
//Cuadrado
char n4[]=("cuadrado");
Cuadrado d("rojo",5,5,n4,15);
cout<<"CUADRADO\n";
d.imprimir();
cout<<"El área es: "<<d.Area()<<endl;
cout<<"El perimetro es: "<<d.Perimetro()<<endl;
cout<<"\n";
```

```
//Circulo
char n5[]=("circulo");
Circulo e("amarillo",-10,-10,n5,10);
cout<<"CIRCULO\n";
e.imprimir();
cout<<"El área es: "<<e.Area()<<endl;
cout<<"\n";
```

```
//Vector de Formas
int n = 4;
Forma** vec = new Forma*[n];
vec[0]=&b;//Rectangulo
vec[1]=&c;//Elipse
vec[2]=&d;//Cuadrado
```

```
vec[3]=&e;//Circulo

//Cambiar color a los objetos en el vector y moverlos
cout<<"CAMBIO DE COLOR Y POSICION PARA LAS 4 FORMAS\n";
for(int i=0; i<n; i++){
    vec[i]->camb_color("café");
    cout<<"vec["<<i<<"]="<<vec[i]->obt_color()<<"\n";
    vec[i]->camb_punto(100,100);
}
cout<<endl;

//Verificar el imprimir()
for(int i=0; i<n; i++){
    cout<<"FORMA EN V["<<i<<"]\n";
    vec[i]->imprimir();
    cout<<endl;
}

//Obtener áreas
cout<<"ÁREAS DE LOS OBJETOS\n";
double ab[]={0,0,0,0};
for(int i=0; i<n; i++){
    ab[i]=vec[i]->Area();
    if(i!=3){
        cout<<ab[i]<<" ";
    }
    else{
        cout<<ab[i]<<endl;
    }
}

//Detecta el area mayor e imprime la informacion del objeto con mayor area
int ptemp=0;
double temp=ab[ptemp];
for(int i=0; i<n; i++){
    if(temp<ab[i]){
        temp=ab[i];
        ptemp=i;
    }
}
cout<<endl;
cout<<"FORMA CON EL ÁREA MAXIMA\n";
vec[ptemp]->imprimir();
cout<<endl;

delete[] vec;

return 0;
}
```

Funcionamiento (Área mayor):

```
FORMA CON EL ÁREA MAXIMA  
El nombre es: elipse  
El color es: café  
La coordenada X es: 100  
La coordenada Y es: 100  
El radio menor es: 20  
El radio mayor es: 40
```

2. Entregables

Al final estudiante deberá:

- 2.1.** Compactar el código elaborado y subirlo al aula virtual de trabajo. Agregue sus datos personales como comentario en cada archivo de código elaborado.
- 2.2.** Elaborar un documento que incluya tanto el código como capturas de pantalla de la ejecución del programa. Este documento debe de estar en formato PDF.
- 2.3.** El nombre del archivo (comprimido como el documento PDF), será su LAB08_GRUPO_A/B/C_CUI_1erNOMBRE_1erAPELLIDO.
(Ejemplo: LAB08_GRUPO_A_2022123_PEDRO_VASQUEZ).
- 2.4.** Debe remitir el documento ejecutable con el siguiente formato:
LAB08_GRUPO_A/B/C_CUI_EJECUTABLE_1erNOMBRE_1erAPELLIDO
(Ejemplo:
LAB08_GRUPO_A_EJECUTABLE_2022123_PEDRO_VASQUEZ).

En caso de encontrarse trabajos similares, los alumnos involucrados no tendrán evaluación y serán sujetos a sanción.