

INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE INGENIERÍA ELÉCTRICA Y MECÁNICA  
UNIDAD ZACATENCO

PROGRAMACIÓN ORIENTADA A OBJETOS

EJERCICIOS DE CASES DEL PRIMER PERIODO DEPARTAMENTAL

CARLOS ALBERTO HERRERA RANGEL

2AM2

# Tabla de contenidos

- 
- EJERCICIOS 08/02/22
  - Ejercicio 1
  - Ejercicio 2
- EJERCICIOS 09/02/22
  - Ejercicio 1
- EJERCICIOS 11/02/22
  - Ejercicio 1
  - Ejercicio 2
- EJERCICIOS 17/02/22
  - Ejercicio 1
- EJERCICIOS 18/02/22
  - Ejercicio 1
  - Ejercicio 2
- EJERCICIOS 22/02/22
  - Ejercicio 1
- Ejercicio 2
- EJERCICIOS 23/02/22
  - Ejercicio 1
  - Ejercicio 2
  - Ejercicio 3
- EJERCICIOS 24/02/22
  - Ejercicio 1
  - Ejercicio 2
- EJERCICIOS 25/02/22
  - Ejercicio 1
  - Ejercicio 2
  - Ejercicio 3
  - Ejercicio 4
- EJERCICIOS 01/03/22
  - Ejercicio 1
  - Ejercicio 2
  - Ejercicio 3
  - Ejercicio 4
- EJERCICIOS 02/03/22
  - Ejercicio 1
  - Ejercicio 2
  - Ejercicio 3

# EJERCICIOS 08/02/22

## Ejercicio 1

Codigo:

```
#include <iostream>
#include <string>
using namespace std;
int main(){
    string nombre, boleta, promedio, profesor, numero, direccion;
    cout<<"Cual es tu nombre? ";
    getline(cin, nombre);
    cout<<"Cual es tu numero de boleta? ";
    getline(cin, boleta);
    cout<<"Cual es tu promedio? ";
    getline(cin, promedio);
    cout<<"Cual fue tu profesor de fundamentos de programacion? ";
    getline(cin, profesor);
    cout<<"Cual es tu numero? ";
    getline(cin, numero);
    cout<<"Cual es tu direccion? ";
    getline(cin, direccion);
    cout<<"Hola "<<nombre<<" Bienvenido al curso de programacion orientada a
objetos!\nBoleta: "<< boleta<<"\nPromedio:"<<promedio<< "\nProfesor:"
<<profesor<<"\nNumero: "<<numero<<"\nDireccion: "<< direccion;
    return 0;
}
```

Ejecutable:

```
$ g++ .\main.cpp -o main; .\main.exe
Cual es tu nombre? Carlos Herrera
Cual es tu numero de boleta? 2022301065
Cual es tu promedio? 8
Cual fue tu profesor de fundamentos de programacion? -
Cual es tu numero? 9212035140
Cual es tu direccion? Coatzacoalcos, Ver
Hola Carlos Herrera Bienvenido al curso de programacion orientada a objetos!
Boleta: 2022301065
Promedio:8
Profesor:-
Numero: 9212035140
Direccion: Coatzacoalcos, Ver
```

## Ejercicio 2

Codigo:

```
#include <iostream>
#include <string>
#define pi 3.1416
using namespace std;
int main(){
    float area, radio;
    cout<<"Ingrese el radio: ";
    cin>>radio;
    area = (radio * radio) * pi;
    cout<<"El area es: "<<area;
    return 0;
}
```

Ejecutable:

```
$ g++ .\main2.cpp -o main2; .\main2.exe
Ingrese el radio: 12
El area es: 452.3
```

# EJERCICIOS 09/02/22

## Ejercicio 1

Codigo:

```
#include <iostream>
using namespace std;
class cuadrado{
    int base, altura;
public:
    void mostrar(int,int);
    int area(){return base * altura;}
};
void cuadrado::mostrar(int h, int b){
    base = b;
    altura = h;
}
int main(){
    cuadrado cuadrado1;
    cuadrado1.mostrar(2,6);
    cout<<"Area: "<<cuadrado1.area()<<endl;
    return 0;
}
```

Ejecutable:

```
$ g++ .\main.cpp -o main; .\main.exe
Area: 12
```

# EJERCICIOS 11/02/22

## Ejercicio 1

Codigo:

```
#include <iostream>
#include <string>
#include <sstream>
using namespace std;
struct pelicula{
    string titulo;
    int fecha;
}mia, suya;
void imprimirPelicula(pelicula miPelicula);
int main(){
    string miString;
    mia.titulo = "interestelar";
    mia.fecha = 2015;
    cout<<"Introduzca el titulo: ";
    cin>>miString;
    stringstream(miString)>>suya.titulo;
    miString = "";
    cout<<"Introduzca la fecha: ";
    cin>>miString;
    stringstream(miString)>>suya.fecha;
    cout<<mia.titulo<<"\n"<<mia.fecha<<"\n"<<suya.titulo<<"\n"<<suya.fecha<<endl;
}
```

Ejecutable:

```
$ g++ .\main.cpp -o main; .\main.exe
Introduzca el titulo: Avatar
Introduzca la fecha: 2009
interestelar
2015
Avatar
2009
```

## Ejercicio 2

Codigo:

```
#include <iostream>
using namespace std;
class rectangulo{
    int altura, largo, ancho;
public:
    void mostrar(int, int, int);
    int volumen(){return altura * largo * ancho;}
};
void rectangulo::mostrar(int h, int l, int w){
    altura = h;
    largo = l;
    ancho = w;
}
int main(){
    rectangulo rec0, rec1, rec2;
    rec0.mostrar(4,4,4);
    rec1.mostrar(5,5,5);
    rec2.mostrar(6,6,6);
    cout<<"El volumen 0 es: "<<rec0.volumen()<<endl<<"El volumen 1 es: "
<<rec1.volumen()<<endl<<"El volumen 2 es: "<<rec2.volumen();
    return 0;
}
```

Ejecutable:

```
$ g++ .\main2.cpp -o main2; .\main2.exe
El volumen 0 es: 64
El volumen 1 es: 125
El volumen 2 es: 216
```

# EJERCICIOS 17/02/22

## Ejercicio 1

Codigo:

```
#include <iostream>
using namespace std;
class poligono{
public:
    int n, l;
    void calcular(int, int);
    int perimetro(){return n * l;}
};
void poligono::calcular(int lado, int numLados){
    n = numLados;
    l = lado;
}
int main()
{
    poligono pol;
    pol.calcular(5,10);
    cout<<"El area es: "<<pol.perimetro();
    return 0;
}
```

Ejecutable:

```
$ g++ .\main.cpp -o main; .\main.exe
El area es: 50
```



# EJERCICIOS 18/02/22

## Ejercicio 1

Codigo:

```
#include <iostream>
using namespace std;
class poligono{
    int perimetro, apotema;
public:
    void calcular(int, int);
    int area(){return perimetro * apotema;};
};
void poligono::calcular(int p, int a){
    perimetro = p;
    apotema = a;
}
int main(){
    poligono pol;
    pol.calcular(67, 6);
    cout<<"El area es: "<<pol.area();
    return 0;
}
```

Ejecutable:

```
$ g++ .\main.cpp -o main; .\main.exe
El area es: 402
```

## Ejercicio 2

Codigo:

```
#include <iostream>
using namespace std;
class poligono{
public:
    int per, apotema, lado, numLados;
    void calcularperimetro(int, int);
    int perimetro(){return lado * numLados;};
    void calcularArea(int,int);
    int area(){return per * apotema;};
};
void poligono::calcularperimetro(int l, int nl){
    lado = l;
    numLados = nl;
}
void poligono::calcularArea(int p, int a){
    per = p;
    apotema = a;
}
int main(){
    poligono pol;
    pol.calcularperimetro(10, 6);
    pol.calcularArea(10,2);
    cout<<"El perimetro es: "<<pol.perimetro()<<endl<<"El area es: "<<pol.area();
    return 0;
}
```

Ejecutable:

```
$ g++ .\main2.cpp -o main2; .\main2.exe
El perimetro es: 60
El area es: 20
```

# EJERCICIOS 22/02/22

## Ejercicio 1

Codigo:

```
#include <iostream>
using namespace std;
class triangulo{
public:
    int lado, altura, per;
    void calcularPerimetro(int);
    int perimetro(){return lado * 3;};
    void calcularArea(int , int);
    int area(){return (lado * altura)/2;};
};
void triangulo::calcularPerimetro(int l){
    lado = l;
}
void triangulo::calcularArea(int l, int h){
    lado = l;
    altura = h;
}
int main(){
    triangulo tri0;
    tri0.calcularPerimetro(3);
    tri0.calcularArea(3, 2);
    cout<<"El perimetro del triangulo es: "<<tri0.perimetro()<<endl<<"El area del
triangulo es: "<<tri0.area()<<endl;
    return 0;
}
```

Ejecutable:

```
$ g++ .\main.cpp -o main; .\main.exe
El perimetro del triangulo es: 9
El area del triangulo es: 3
```

## Ejercicio 2

Codigo:

```
#include <iostream>
using namespace std;
class Cubo{
    int lado;
    public:
    void valor(int);
    int volumen(){return lado * lado * lado;}
};
void Cubo::valor(int l){
    lado=l;
}
int main(){
    Cubo cub;
    cub.valor(3);
    cout<<cub.volumen()<<endl;
}
```

Ejecutable:

```
$ g++ .\main2.cpp -o main2; .\main2.exe
27
```

# EJERCICIOS 23/02/22

## Ejercicio 1

Codigo:

```
#include <iostream>
using namespace std;
class Triangulo{
    int base, altura;
public:
    void calcularArea(int, int);
    int area(){return(base * altura)/2;};
}tria0, tria1;
void Triangulo::calcularArea(int b, int h){
    base = b;
    altura = h;
}
int main(){
    tria0.calcularArea(2, 8);
    tria1.calcularArea(3, 8);
    cout<<"Area del triangulo 1: "<<tria0.area()<<endl<<"Area de triangulo 2: "
<<tria1.area()<<endl;
    return 0;
}
```

Ejecutable:

```
$ g++ .\main.cpp -o main; .\main.exe
Area del triangulo 1: 8
Area de triangulo 2: 12
```

## Ejercicio 2

Codigo:

```
#include <iostream>
using namespace std;
class Rectangulo{
    int base, altura;
public:
    Rectangulo();
    Rectangulo(int, int);
    int area(void){return base*altura;};
};
Rectangulo::Rectangulo(){
    base = 2;
    altura = 3;
}
Rectangulo::Rectangulo(int b, int h){
    base = b;
    altura = h;
}
int main(){
    Rectangulo rect(4, 5);
    Rectangulo rect2;
    cout<<"Area 1: "<<rect.area()<<endl<<"Area 2: "<<rect2.area()<<endl;
    return 0;
}
```

Ejecutable:

```
$ g++ .\main2.cpp -o main2; .\main2.exe
Area 1: 20
Area 2: 6
```

## Ejercicio 3

Codigo:

```
#include <iostream>
using namespace std;
class Circulo{
    double radio;
public:
    Circulo(double r){radio = r;};
    double circumferencia(){return 2* radio * 3.1416;};
};
int main(){
    Circulo circ1 = 20;
    Circulo circ(10.0);
    Circulo circ2 (20);
    Circulo circ3 = (10);
    cout<<"Circunferencia: "<<circ.circumferencia()<<endl;
    cout<<"Circunferencia: "<<circ1.circumferencia()<<endl;
    cout<<"Circunferencia: "<<circ2.circumferencia()<<endl;
    cout<<"Circunferencia: "<<circ3.circumferencia()<<endl;
    return 0;
}
```

Ejecutable:

```
$ g++ .\main3.cpp -o main3; .\main3.exe
Circunferencia: 62.832
Circunferencia: 125.664
Circunferencia: 125.664
Circunferencia: 62.832
```

# EJERCICIOS 24/02/22

## Ejercicio 1

Codigo:

```
#include <iostream>
using namespace std;
class Figura{
    int lado, numLados, per, apotema;
public:
    Figura(int l, int nl, int p, int a){lado = l, numLados = nl, per = p, apotema =
a;}
    int perimetro(){return lado * numLados;}
    int area(){return per * apotema;}
};
int main(){
    Figura pentagono(10, 5, 50, 6);
    cout<<"Perimetro: "<<pentagono.perimetro()<<endl<<"Area: "<<pentagono.area();
}
```

Ejecutable:

```
$ g++ .\main.cpp -o main; .\main.exe
Perimetro: 50
Area: 300
```



## Ejercicio 2

Codigo:

```
#include <iostream>
#include <string>
using namespace std;
class Figura{
    int numLados;
    double longLado, per, apotema, altura;
    string tipo;
public:
    Figura();
    Figura(double, int, double, double, double);
    double perimetro(void);
    double area(void);
};
Figura::Figura(){
    longLado = 0;
    numLados = 0;
    per = 0;
    apotema = 0;
    altura = 0;
}
Figura::Figura(double l, int nl, double p, double a, double h){
    longLado = l;
    numLados = nl;
    per = p;
    apotema = a;
    altura = h;
    if (nl == 3){
        tipo = "triangulo";
    }
    else if (nl == 4){
        tipo = "cuadrado";
    }
    else if (nl > 4){
        tipo = "poligono";
    }
    else{
        tipo = "invalido";
    }
}
double Figura::perimetro(void){
    if (tipo == "invalido"){
        return 0;
    }
}
```

```

        else{
            return longLado * numLados;
        }
    }
}

double Figura::area(void){
    if (tipo == "triangulo"){
        return (longLado * altura) / 2;
    }
    else if (tipo == "cuadrado"){
        return longLado * longLado;
    }
    else if (tipo == "poligono"){
        return per * apotema;
    }
    else{
        return 0;
    }
}

int main(){
    double l, p, a, h;
    int nl;
    cout<<"Longitud del lado: " << endl;
    cin>>l;
    cout<<"Numero de lados: " << endl;
    cin>>nl;
    cout<<"Perimetro: " << endl;
    cin>>p;
    cout<<"Apotema: " << endl;
    cin>>a;
    cout<<"Altura: " << endl;
    cin>>h;
    Figura figura(l, nl, p, a, h);
    Figura triagulo (3, 3, 9, 0, 4);
    Figura cuadrado (2, 4, 8, 0, 2);
    Figura pentagono (3, 5, 15, 4, 8);
    cout<<"Figura:"<<endl<<"Perimetro: "<<figura.perimetro()<<endl<<"Area: "
<<figura.area()<<endl<<endl;
    cout<<"TRIANGULO:"<<endl<<"Perimetro: "<<triagulo.perimetro()<<endl<<"Area: "
<<triagulo.area()<<endl<<endl;
    cout<<"CUADRADO:"<<endl<<"Perimetro: "<<cuadrado.perimetro()<<endl<<"Area: "
<<cuadrado.area()<<endl<<endl;
    cout<<"POLIGONO:"<<endl<<"Perimetro: "<<pentagono.perimetro()<<endl<<"Area: "
<<pentagono.area()<<endl<<endl;
    return 0;
}

```

## Ejecutable:

```
$ g++ .\main2.cpp -o main2; .\main2.exe
Longitud del lado:
20
Numero de lados:
4
Perimetro:
2
Apotema:
3
Altura:
3
Figura:
Perimetro: 80
Area: 400

TRIANGULO:
Perimetro: 9
Area: 6

CUADRADO:
Perimetro: 8
Area: 4

POLIGONO:
Perimetro: 15
Area: 60
```

# EJERCICIOS 25/02/22

## Ejercicio 1

Codigo:

```
#include <iostream>
using namespace std;
class Rectangulo{
    int base, altura;
public:
    Rectangulo(int b, int h):base(b), altura(h){};
    Rectangulo() = default;
    Rectangulo(const Rectangulo & other) = delete;
    int area(){return base * altura;};
};
int main(){
    Rectangulo rec(6,7);
    Rectangulo rec1(5,7);
    cout<<"Area: "<<rec.area()<<endl;
    cout<<"Area 2 : "<<rec1.area()<<endl;
    return 0;
}
```

Ejecutable:

```
$ g++ .\main.cpp -o main; .\main.exe
Area: 42
Area 2 : 35
```

## Ejercicio 2

Codigo:

```
#include <iostream>
using namespace std;
class demostracion{
public:
    demostracion();
    ~demostracion();
};
demostracion::demostracion(){
    cout<<"Se acaba de definir un object, el constructor esta corriendo" << endl;
}
demostracion::~demostracion(){
    cout<<"El destructor esta corriendo"<<endl;
}
int main(){
    demostracion demo;
    cout<<"El object esta apunto de ser destruido"<<endl;
    return 0;
}
```

Ejecutable:

```
$ g++ .\main2.cpp -o main2; .\main2.exe
Se acaba de definir un object, el constructor esta corriendo
El object esta apunto de ser destruido
El destructor esta corriendo
```

## Ejercicio 3

Codigo:

```
#include <iostream>
using namespace std;
class Mascota{
public:
virtual ~Mascota() = 0;
};
Mascota::~Mascota(){
    cout<<"~Mascota()"<<endl;
}
class Perro : public Mascota{
public:
    ~Perro(){
        cout<<"~Perro()"<<endl;
    }
};
int main(){
    Mascota *p = new Perro;
    delete p;
    return 0;
}
```

Ejecutable:

```
$ g++ .\main3.cpp -o main3; .\main3.exe
~Perro()
~Mascota()
```

## Ejercicio 4

Codigo:

```
#include <iostream>
using namespace std;
class Ficticia{
public:
    static int n;
    Ficticia(){n++;}
};
int Ficticia::n = 0;
int main(){
    Ficticia a;
    Ficticia b[5];
    cout << a.n << endl;
    Ficticia *c = new Ficticia;
    cout << Ficticia::n << endl;
    delete c;
}
```

Ejecutable:

```
$ g++ .\main4.cpp -o main4; .\main4.exe
6
7
```

# EJERCICIOS 01/03/22

## Ejercicio 1

Codigo:

```
#include <iostream>
using namespace std;
int operacion(int a, int b){
    return a * b;
}
double operacion(double a, double b){
    return a/b;
}
int main(){
    int x = 5, y = 2;
    double n = 5.0, m = 2.0;
    cout<<operacion(x,y)<<endl;
    cout<<operacion(n, m)<<endl;
    return 0;
}
```

Ejecutable:

```
$ g++ .\main.cpp -o main; .\main.exe
10
2.5
```



## Ejercicio 2

Codigo:

```
#include <iostream>
using namespace std;
class Circunferencia{
    double radio;
public:
    Circunferencia(double r){radio = r;}
    double area(){return (radio * radio) * 3.1416;}
    double perimetro(){return 2 * radio * 3.1416;}
    double volumen(){return 4/3*(3.1416)*(radio * radio * radio);}
};
int main(){
    Circunferencia cir0(10.0);
    cout<<"Area: "<<cir0.area()<<endl<<"Perimetro: "<<cir0.perimetro()
<<endl<<"Volumen: "<<cir0.volumen()<<endl;
    return 0;
}
```

Ejecutable:

```
$ g++ .\main2.cpp -o main2; .\main2.exe
Area: 314.16
Perimetro: 62.832
Volumen: 3141.6
```

## Ejercicio 3

Codigo:

```
#include<iostream>
using namespace std;
class Rec{
    int base, altura, ancho;
public:
    Rec(int b, int h, int a){base = b, altura = h, ancho = a;}
    ~Rec(){cout << "Objeto destuido";}
    int area(){return base * altura;}
    int perimetro(){return 2 * (base) + 2 * (altura);}
    int volumen(){return base * altura * ancho;}
};
int main(){
    Rec *rec;
    rec = new Rec(2, 4, 6);
    cout<<"Area: "<<rec->area()<<endl<<"Perimetro: "<<rec->perimetro()
<<endl<<"Volumen: "<<rec->volumen()<<endl;
    delete rec;
    return 0;
}
```

Ejecutable:

```
$ g++ .\main3.cpp -o main3; .\main3.exe
Area: 8
Perimetro: 12
Volumen: 48
Objeto destuido
```

## Ejercicio 4

Codigo:

```
#include<iostream>
using namespace std;
class Rec{
    int base, altura, ancho;
public:
    Rec(int b, int h, int a){base = b, altura = h, ancho = a;}
    ~Rec(){cout << "Objeto destruido \n";}
    int area(){return base * altura;}
    int perimetro(){return 2 * (base) + 2 * (altura);}
    int volumen(){return base * altura * ancho;}
};
int main(){
    Rec obj(2,3,4);
    Rec *foo, *bar, *baz;
    foo = &obj;
    bar = new Rec(1, 2, 3);
    baz = new Rec[2]{{4, 5, 6},{7, 8, 9}};
    cout<<"AREAS: " <<endl<<"Foo: " <<foo->area()<<"\t"<<"Bar: " <<bar->area()<<"\t"
    <<"Baz: " <<baz->area()<<endl;
    cout<<"PERIMETRO: " <<endl<<"Foo: " <<foo->perimetro()<<"\t"<<"Bar: " <<bar-
    >perimetro()<<"\t"<<"Baz: " <<baz->perimetro()<<endl;
    cout<<"VOLUMEN: " <<endl<<"Foo: " <<foo->volumen()<<"\t"<<"Bar: " <<bar->volumen()
    <<"\t"<<"Baz: " <<baz->volumen()<<endl;
    foo->~Rec();
    bar->~Rec();
    baz->~Rec();
    return 0;
}
```

Ejecutable:

```
$ g++ .\main4.cpp -o main4; .\main4.exe
AREAS:
Foo: 6 Bar: 2 Baz: 20
PERIMETRO:
Foo: 10 Bar: 6 Baz: 18
VOLUMEN:
Foo: 24 Bar: 6 Baz: 120
Objeto destruido
Objeto destruido
Objeto destruido
Objeto destruido
```

# EJERCICIOS 02/03/22

## Ejercicio 1

Codigo:

```
#include <iostream>
#include <string>
using namespace std;
class ejemplo{
    string *ptr;
public:
    ejemplo():ptr(new string){}
    ejemplo(const string& str):ptr(new string(str)){}
    ~ejemplo(){delete ptr;}
    const string& contenido(){return *ptr;}
};
int main(){
    ejemplo foo("examen");
    ejemplo bar("examen");
    cout<<"foo contenido: "<<foo.contenido()<<endl;
    cout<<"bar contenido: "<<bar.contenido()<<endl;
    return 0;
}
```

Ejecutable:

```
$ g++ .\main.cpp -o main; .\main.exe
foo contenido: examen
bar contenido: examen
```

## Ejercicio 2

Codigo:

```
#include <iostream>
using namespace std;
class delfines{
public:
    virtual ~delfines()=0;
};
delfines::~delfines(){
    cout<<"~cetaceos"<<endl;
}
class distintosdelfines:public delfines{
public:
    ~distintosdelfines(){
        cout<<"~delfin()"<<endl;}
};
int main(){
    delfines *p=new distintosdelfines;
    delete p;
}
```

Ejecutable:

```
$ g++ .\main2.cpp -o main2; .\main2.exe
~delfin()
~cetaceos
```

## Ejercicio 3

Codigo:

```
#include <iostream>
#include <string>
#include <cstdlib>
using namespace std;
class Ali {
public:
    int *imprimocomando;
    int miAli;
    Ali(){
        cout<<"Destructor ~ Ali llamado:"<<endl;
    }
    Ali (int miAlic):miAli(miAlic){
        cout<<"Constructor Ali ("<<miAlic<<") llamado"<<endl;
        imprimocomando = new int[1000];
        for (int i=0; i < miAlic;i++) imprimocomando[i] = miAlic *1000 + i;
    }
    ~Ali(){
        cout<<"Destructor ~ Ali llamado"<<endl;
        delete [] imprimocomando;
    }
};
int main (int argc, const char *argv[]){
    Ali t;
    t = Ali(5);
    cout<<"Investiga la llamada inesperada de un destructor de la clase Ali:";
    for (int i=0; i < t.miAli; i++) cout<<" "<<t.imprimocomando[i];
    cout<<endl;
    return 0;
}
```

Ejecutable:

```
$ g++ .\main3.cpp -o main3; .\main3.exe
Destructor ~ Ali llamado:
Constructor Ali (5) llamado
Destructor ~ Ali llamado
Investiga la llamada inesperada de un destructor de la clase Ali: 16711872 16741472
5002 5003 5004
Destructor ~ Ali llamado
```