

Laboratorio 06

Clases

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. Implemente un programa con clases que calcule el área de un rectángulo y perímetro.

Código (“rectangle.h”):

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

class rectangle{
public:
    rectangle(double,double);
    ~rectangle();
    void height_Value(double);
    void wide_Value(double);
    void Area();
    void Perimeter();
private:
    double height;
    double wide;
};

#endif
```

Código (“rectangle.cpp”):

```
#include "rectangle.h"

rectangle::rectangle(double _height, double _wide){
```

```
        height=abs(_height);
        wide=abs(_wide);
    }
    rectangle::~rectangle(){
    }
    void rectangle::height_Value(double _height){
        height=abs(_height);
    }
    void rectangle::wide_Value(double _wide){
        wide=abs(_wide);
    }
    void rectangle::Area(){
        double area = height * wide;
        cout<<"El area del rectangulo es: "<<area<<"\n";
    }
    void rectangle::Perimeter(){
        double perimeter = (2*height) + (2*wide);
        cout<<"El perimetro del rectangulo es: "<<perimeter<<"\n";
    }
}
```

Código (“LAB_06.cpp”):

```
#include <iostream>
#include "rectangle.h"
using namespace std;

int main(){
    //Clase Rectangulo
    rectangle one(10,20);
    //Primer Area y Perimetro del rectangulo
    cout<<"Primer Rectangulo\n";
    one.Area();
    one.Perimeter();
    //Cambiar valores del rectangulo
    one.height_Value(15);
    one.wide_Value(35);
    //Segunda Area y Perimetro del rectangulo
    cout<<"Segundo Rectangulo\n";
    one.Area();
    one.Perimeter();

    return 0;
}
```

Funcionalidad:

```
Primer Rectangulo
El area del rectangulo es: 200
El perimetro del rectangulo es: 60
Segundo Rectangulo
El area del rectangulo es: 525
El perimetro del rectangulo es: 100
```

2. Se conoce de un alumno de la Universidad Nacional de San Agustín: CUI, nombre completo y tres notas parciales (nota1, nota2, nota3). El programa con clases debe imprimir: CUI, el primer nombre, el promedio de las tres notas e indique con un mensaje si el alumno aprobó (nota final ≥ 10.5) o no aprobó (nota final < 10.5) la asignatura de Ciencias de la Computación II.

Código (“student.h”):

```
#ifndef STUDENT_H
#define STUDENT_H
#include <string>
#include <iostream>
#include <cmath>
using namespace std;

class student {
public:
    student(int,string,float,float,float);
    ~student();
    void p_CUI();
    void pf_name();
    void p_average_score();

private:
    int CUI;
    string full_name;
    float score1;
    float score2;
    float score3;
};

#endif
```

Código (“student.cpp”):

```
#include "student.h"

student::student(int _CUI, string _full_name, float _score1, float _score2, float _score3) {
    CUI=_CUI;
    full_name=_full_name;
    score1=abs(_score1);
    score2=abs(_score2);
    score3=abs(_score3);
}

student::~student(){
}

void student::p_CUI(){
    cout<<"El CUI del estudiante es: "<<CUI<<"\n";
}

void student::pf_name(){
    cout<<"El nombre del estudiante es: "<<full_name<<"\n";
}

void student::p_average_score(){
```

```
float temp=(score1+score2+score3)/3;
cout<<"La nota promedio del estudiante es: "<<temp<<"\n";
if(temp>=10.5){
    cout<<"El estudiante SI ha aprobado Ciencias de la Computacion II\n";
}
else{
    cout<<"El estudiante NO ha aprobado Ciencias de la Computacion II\n";
}
}
```

Código (“LAB_06.cpp”):

```
#include <iostream>
#include "student.h"
using namespace std;

int main(){

    student marcos(20221213,"Marcos Perez",10.5,10.5,10.5);
    marcos.pf_name();
    marcos.p_CUI();
    marcos.p_average_score();

    return 0;
}
```

Funcionalidad:

```
El nombre del estudiante es: Marcos Perez
El CUI del estudiante es: 20221213
La nota promedio del estudiante es: 10.5
El estudiante SI ha aprobado Ciencias de la Computacion II
```

3. Implemente un programa con clases que lea la fecha de nacimiento y la fecha de hoy y muestre por pantalla el nombre y la edad de la persona.

Código (“person.h”):

```
#ifndef PERSON_H
#define PERSON_H
#include <string>
#include <iostream>
#include <cstdlib>
using namespace std;

class person{
public:
    person(string,string,string);
}
```

```
~person();  
void let_dateOB(string);  
void let_dateON(string);  
void show_Name();  
void actual_Age();  
private:  
    string name;  
    string bird;  
    string now;  
};  
  
#endif
```

Código (“person.cpp”):

```
#include "person.h"  
  
person::person(string _name, string _bird, string _now){  
    name=_name;  
    bird=_bird;  
    now=_now;  
}  
person::~~person(){  
}  
void person::let_dateOB(string _bird){  
    bird=_bird;  
}  
void person::let_dateON(string _now){  
    now=_now;  
}  
void person::show_Name(){  
    cout<<"El nombre de la persona es: "<<name;  
}  
void person::actual_Age(){  
    string temp;  
    string actual;  
    //Fecha de Nacimiento  
    int yx;  
    int mx;  
    int dx;  
  
    //Fecha Actual  
    int yy;  
    int my;  
    int dy;  
  
    //Edad Actual  
    int yz=0;  
    int mz=1;  
    int dz=1;  
  
    //Separación Nacimiento  
    temp=bird.substr(0, 4);  
    yx=stoi(temp);  
    temp=bird.substr(5, 2);
```

```
mx=stoi(temp);
temp=bird.substr(8, 2);
dx=stoi(temp);
```

```
//Separación Fecha Actual
temp=now.substr(0, 4);
yy=stoi(temp);
temp=now.substr(5, 2);
my=stoi(temp);
temp=now.substr(8, 2);
dy=stoi(temp);
```

```
//Calculo de Edad Actual en años, meses y días
```

```
while((yx<yy)||((mx<my)||((dx<dy))){
    dx++;
    if(mx==1&&dx==32){
        mx++;
        dx=1;
    }
    if(mx==2&&dx==30){
        mx++;
        dx=1;
    }
    if(mx==2&&dx==29){
        mx++;
        dx=1;
        if(yx%4==0){
            mx--;
            dx=29;
        }
    }
    if(mx==3&&dx==32){
        mx++;
        dx=1;
    }
    if(mx==4&&dx==31){
        mx++;
        dx=1;
    }
    if(mx==5&&dx==32){
        mx++;
        dx=1;
    }
    if(mx==6&&dx==31){
        mx++;
        dx=1;
    }
    if(mx==7&&dx==32){
        mx++;
        dx=1;
    }
    if(mx==8&&dx==32){
        mx++;
        dx=1;
    }
    if(mx==9&&dx==31){
```

```
        mx++;
        dx=1;
    }
    if(mx==10&&dx==32){
        mx++;
        dx=1;
    }
    if(mx==11&&dx==31){
        mx++;
        dx=1;
    }
    if(mx==12&&dx==32){
        mx=1;
        dx=1;
        yx++;
    }
    dz++;
    if(mz==1&&dz==32){
        mz++;
        dz=1;
    }
    if(mz==2&&dz==30){
        mz++;
        dz=1;
    }
    if(mz==2&&dz==29){
        mz++;
        dz=1;
        if(yz%4==0){
            mz--;
            dz=29;
        }
    }
    if(mz==3&&dz==32){
        mz++;
        dz=1;
    }
    if(mz==4&&dz==31){
        mz++;
        dz=1;
    }
    if(mz==5&&dz==32){
        mz++;
        dz=1;
    }
    if(mz==6&&dz==31){
        mz++;
        dz=1;
    }
    if(mz==7&&dz==32){
        mz++;
        dz=1;
    }
    if(mz==8&&dz==32){
        mz++;
        dz=1;
    }
```

```
    }
    if(mz==9&&dz==31){
        mz++;
        dz=1;
    }
    if(mz==10&&dz==32){
        mz++;
        dz=1;
    }
    if(mz==11&&dz==31){
        mz++;
        dz=1;
    }
    if(mz==12&&dz==32){
        mz=1;
        dz=1;
        yz++;
    }
}

//Transforma ints a string
temp=to_string(yz);
actual=temp;
temp=to_string(mz);
actual+="-";
actual+=temp;
temp=to_string(dz);
actual+="-";
actual+=temp;

//Edad actual
cout<<"Su edad actual seria (formato Y-M-D): "<<actual;
}
```

Código (“LAB_06.cpp”):

```
#include <iostream>
#include "person.h"
using namespace std;

int main(){
    //Ingresar las fechas en formato YYYY-MM-DD
    person pepe("Pepe","2002-01-23","2022-05-12");
    pepe.show_Name();
    cout<<"\n";
    pepe.actual_Age();
    cout<<"\n";

    pepe.let_dateOB("2003-05-26");
    pepe.let_dateON("2022-05-26");
    pepe.show_Name();
    cout<<"\n";
    pepe.actual_Age();
    cout<<"\n";
}
```



```
    return 0;  
}
```

Funcionalidad:

```
El nombre de la persona es: Pepe  
Su edad actual seria (formato Y-M-D): 20-4-19  
El nombre de la persona es: Pepe  
Su edad actual seria (formato Y-M-D): 19-1-1
```

4. Implemente una aplicación con clases donde una clase contiene un arreglo en el que se pueden almacenar como máximo 5 datos. Debe tener un nuevo dato para almacenarlo, debe existir un método que verifique que el arreglo tenga espacio disponible para guardar el dato. En caso contrario se debe mostrar un mensaje indicando que el arreglo está lleno. Asimismo, debe tener un método para retirar un dato del arreglo, este método debe verificar que, si haya algo para sacar del arreglo, es decir que el arreglo no vaya a estar vacío, en cuyo caso debe desplegar un mensaje diciendo que no hay nada para sacar de arreglo.

Código (“array5v.h”):

```
#ifndef ARRAY5V_H  
#define ARRAY5V_H  
#include <iostream>  
using namespace std;  
  
class array5v {  
public:  
    array5v();  
    ~array5v();  
    bool if_Space();  
    bool somethingTo_Remove();  
    void new_Element(int);  
    void remove_Element(int);  
    void show_Array();  
private:  
    int** arr5;  
    int p_it=0;  
};  
  
#endif
```

Código (“array5v.cpp”):

```
#include "array5v.h"
```

```
array5v::array5v(){
    arr5=new int* [5];
    for(int i=0; i<5; i++){
        arr5[i]=NULL;
    }
}
array5v::~~array5v(){
    for(int i=0; i<5; i++){
        delete arr5[i];
    }
    delete[] arr5;
}
bool array5v::if_Space(){
    if(p_it==5){
        return false;
    }
    return true;
}
bool array5v::somethingTo_Remove(){
    bool* barr5=new bool [5];
    for(int i=0; i<5; i++){
        if(arr5[i]==NULL){
            barr5[i]=false;
        }
        else{
            barr5[i]=true;
        }
    }
    for(int i=0; i<5; i++){
        if(barr5[i]==true){
            delete[] barr5;
            return true;
        }
    }
    delete[] barr5;
    return false;
}
void array5v::new_Element(int a){
    if(if_Space()==true){
        arr5[p_it]=new int(a);
        cout<<"Se agrego exitosamente el valor \""<<a<<"\" en la posicion \""<<p_it<<"\" del
array\n";
        p_it++;
    }
    else{
        cout<<"El array esta completamente lleno\n";
    }
}
void array5v::remove_Element(int a){
    if(somethingTo_Remove()==true){
        for(int i=0; i<5; i++){
            if(*arr5[i]==a){
                delete arr5[i];
                arr5[i]=NULL;
                if(arr5[i+1]!=NULL){
                    for(int j=i; j<4; j++){
```

```
        arr5[j]=arr5[j+1];
        arr5[j+1]=NULL;
    }
    p_it--;
}
cout<<"Se elimino exitosamente el valor \""<a<<" en la posicion
\"<i<<" del array\n";
    break;
}
}
else{
    cout<<"No hay ningun elemento para eliminar\n";
}
}
void array5v::show_Array(){
    cout<<"[";
    for(int i=0; i<5; i++){
        if(i<=3&&arr5[i]!=NULL){
            cout<<*arr5[i]<<" ";
        }
        else if(i<=3&&arr5[i]==NULL){
            cout<<"NULL, ";
        }
        else if(i>3&&arr5[i]!=NULL){
            cout<<*arr5[i]<<"]";
        }
        else if(i>3&&arr5[i]==NULL){
            cout<<"NULLL]";
        }
    }
    cout<<"\n";
}
```

Código (“LAB_06.cpp”):

```
#include <iostream>
#include "array5v.h"
using namespace std;

int main(){
    array5v arrx;
    cout<<"PRIMER ARRAY\n";
    arrx.new_Element(10);
    arrx.new_Element(5);
    arrx.new_Element(2);
    arrx.new_Element(3);
    arrx.new_Element(2);
    arrx.show_Array();
    arrx.new_Element(12);
    cout<<"\n";

    cout<<"PRIMER ARRAY (PRIMERAS MODIFICACIONES)\n";
    arrx.remove_Element(2);
    arrx.show_Array();
}
```

```
arrx.remove_Element(5);
arrx.show_Array();
arrx.new_Element(11);
arrx.show_Array();
cout<<"\n";

cout<<"PRIMER ARRAY (SEGUNDAS MODIFICACIONES)\n";
arrx.remove_Element(11);
arrx.remove_Element(2);
arrx.remove_Element(3);
arrx.remove_Element(10);
arrx.show_Array();
arrx.remove_Element(5);

return 0;
}
```

Funcionamiento:

```
PRIMER ARRAY
Se agrego exitosamente el valor "10" en la posicion "0" del array
Se agrego exitosamente el valor "5" en la posicion "1" del array
Se agrego exitosamente el valor "2" en la posicion "2" del array
Se agrego exitosamente el valor "3" en la posicion "3" del array
Se agrego exitosamente el valor "2" en la posicion "4" del array
[10, 5, 2, 3, 2]
El array esta completamente lleno

PRIMER ARRAY (PRIMERAS MODIFICACIONES)
Se elimino exitosamente el valor "2" en la posicion "2" del array
[10, 5, 3, 2, NULLL]
Se elimino exitosamente el valor "5" en la posicion "1" del array
[10, 3, 2, NULL, NULLL]
Se agrego exitosamente el valor "11" en la posicion "3" del array
[10, 3, 2, 11, NULLL]

PRIMER ARRAY (SEGUNDAS MODIFICACIONES)
Se elimino exitosamente el valor "11" en la posicion "3" del array
Se elimino exitosamente el valor "2" en la posicion "2" del array
Se elimino exitosamente el valor "3" en la posicion "1" del array
Se elimino exitosamente el valor "10" en la posicion "0" del array
[NULL, NULL, NULL, NULL, NULLL]
No hay ningun elemento para eliminar
```

5. Implementar un programa con clases que haga la búsqueda de un dato almacenado en una matriz de 3x3 que tiene los números de 1 al 9 ingresados aleatoriamente y debe indicar la posición donde se encuentra el dato.

Código (“mat3x3.h”):

```
#ifndef MAT3X3_H
#define MAT3X3_H
#include <iostream>
#include <ctime>
using namespace std;

class mat3x3 {
public:
    bool verifi_Random(int*,int,int&);
    mat3x3();
    ~mat3x3();
    void show_Mat();
    void search_Val(int);
private:
    int mat[3][3];
};

#endif
```

Código (“mat3x3.cpp”):

```
#include "mat3x3.h"

bool mat3x3::verifi_Random(int* vec, int n, int &it){
    for(int i=0; i<9; i++){
        if(n==vec[i]){
            return false;
        }
    }
    vec[it]=n;
    it++;
    return true;
}

mat3x3::mat3x3(){
    int* vec=new int[9];
    fill(vec,vec+9,0);
    int it=0;

    srand(time(NULL));
    for(int i=0; i<3; i++){
        for(int j=0; j<3; j++){
            int n = rand() % 9 + 1;
            if(verifi_Random(vec,n,it)==true){
                mat[i][j]=n;
            }
            else{

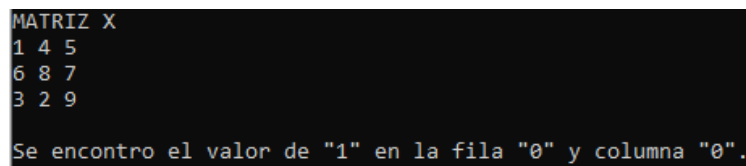
```

```
        j--;\n    }\n\n    delete[] vec;\n}\n\nmat3x3::~~mat3x3()\n{\n}\n\nvoid mat3x3::show_Mat()\n{\n    for(int i=0; i<3; i++){\n        for(int j=0; j<3; j++){\n            cout<<mat[i][j]<<" "; \n\n        }\n        cout<<"\\n";\n    }\n    cout<<"\\n";\n}\n\nvoid mat3x3::search_Val(int val){\n    for(int i=0; i<3; i++){\n        for(int j=0; j<3; j++){\n            if(mat[i][j]==val){\n                cout<<"Se encontro el valor de \""<<val<<"\" en la fila \""<<i<<"\" y\n                columna \""<<j<<"\"."; \n            }\n        }\n    }\n}\n\n}
```

Código ("LAB_06.cpp"):

```
#include <iostream>\n#include "mat3x3.h"\nusing namespace std;\n\nint main(){\n    mat3x3 x;\n    cout<<"MATRIZ X\\n";\n    x.show_Mat();\n    x.search_Val(1);\n\n    return 0;\n}
```

Funcionamiento:



```
MATRIZ X\n1 4 5\n6 8 7\n3 2 9\n\nSe encontro el valor de "1" en la fila "0" y columna "0".
```

2. Entregables

Al final estudiante deberá:

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. 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.
3. El nombre del archivo (comprimido como el documento PDF), será su LAB06_GRUPO_A/B/C_CUI_1erNOMBRE_1erAPELLIDO.

(Ejemplo: LAB06_GRUPO_A_2022123_PEDRO_VASQUEZ).

4. Debe remitir el documento ejecutable con el siguiente formato:

LAB06_GRUPO_A/B/C_CUI_EJECUTABLE_1erNOMBRE_1erAPELLIDO

(Ejemplo: LAB06_GRUPO_A_EJECUTABLE_2022123_PEDRO_VASQUEZ).

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