UNIVERSIDAD NACIONAL DE INGENIERÍA

Facultad de Ciencias
Escuela Profesional de Ciencia de la Computación

Curso: Fundamentos de Programación CC112 Semestre 2025-I

Laboratorio 14

Temas

Clases, constructores, destructores, programas orientado a objetos

1. Defina una clase con dos variables enteras privadas y dos funciones miembro: una para ingresar los valores desde el teclado y otra para mostrarlos en pantalla.
2. Defina una clase `Estudiante` con tres atributos: `nombre`, `edad` y `ciclo`. Implemente dos funciones miembro: una para ingresar los datos y otra para mostrarlos. Cree dos objetos y utilice estas funciones para operar con ellos.
3. Usando clases, resuelva los siguientes problemas:
   - Calcular el factorial de un número.
   - Invertir un número entero (Ejemplo: Entrada: 1234, Salida: 4321).
   - Verificar si un número o palabra es palíndroma (Ejemplo: 313, 3223).
   - Verificar si un número es perfecto (Ejemplo: 6 = 1 + 2 + 3).
   - Verificar si un número es de Armstrong.
   - Generar los primeros `n` términos de la sucesión de Fibonacci.
4. Defina una clase con un miembro de datos de tipo `enum`. El tipo `enum` es una enumeración definida por el usuario que contiene constantes integrales. Ejemplo:

   ```
   enum Estaciones {primavera, verano, otoño, invierno};
   ```

   Por defecto, los valores son 0, 1, 2 y 3 respectivamente. Estos valores pueden redefinirse si se desea.
5. Cree una clase `Persona` cuyos miembros de datos incluyan: `nombre`, `edad`, `salario` y una estructura llamada `Direccion` con los atributos: `hno` (número de casa), `calle`, `ciudad` y `estado`. Implemente funciones miembro para ingresar y mostrar los datos.
6. Implemente una clase con miembros de datos privados. Cree una función miembro que permita acceder a dichos miembros desde fuera de la clase.
7. Dada una matriz cuadrada de orden n, implemente usando clases:

a) Un método para ordenar los elementos por filas.

b) Un método para imprimir la suma de los elementos de la diagonal principal.

8. Escriba un programa en C++ utilizando clases que permita convertir letras mayúsculas a minúsculas y viceversa en una cadena de caracteres.

9. Implemente una clase que permita invertir una cadena de caracteres. La inversión se debe realizar mediante una función miembro que utilice asignación de memoria dinámica.

10. Implemente una clase para determinar si una cadena es palíndroma. Ejemplos: "sos", "peep".

11. Cree una clase `Empleado` con miembros `nombre` y `salario`. Implemente una función miembro para ingresar datos y otra que reciba un objeto de tipo `Empleado` y compare los salarios.

12. Implemente una clase `Time` con tres miembros: `horas`, `minutos`, `segundos`. Incluya dos funciones miembro:

    - `input_time(int hh, int mm, int ss)`: para ingresar datos.
    - `comp_time(Time)`: para comparar dos objetos de tiempo.

13. Cree una clase con un dato entero y una función miembro que devuelva un objeto de esa clase.

14. Cree una clase `Time` para calcular la suma de dos tiempos. Por ejemplo, si:

    Tiempo1 = 3:35:45
    Tiempo2 = 4:56:45
    Resultado: 8:32:30

    Use una función miembro con prototipo: `Time sum_time(Time, Time);`.

15. Utilice un arreglo de objetos para representar las coordenadas de 10 puntos en el plano y mostrarlas en pantalla.

16. Desarrolle un programa que use clases y un arreglo de objetos para leer y mostrar el nombre y salario de los empleados de una empresa. Ejemplo de salida:

Nombre	**Salario**
Abel	2600
Paul	2000
Raquel	2100

17. Defina una clase con un dato entero y tres funciones miembro: una para inicializar el dato, otra para cambiarlo (debe ser constante), y otra para mostrarlo.

18. Escriba un programa que utilice clases para calcular el factorial de un número, usando un constructor por defecto.

19. Implemente un programa que, dado un número con parte decimal (Ejemplo: 234.23), permita separar su parte entera y su parte decimal.

20. Cree una clase para sumar y restar dos números complejos. La clase debe manejar los datos reales e imaginarios.

21. Convierta una matriz de fracciones a su forma escalonada. Para ello, implemente las siguientes clases y métodos:

    a) Defina una clase `Fraccion`, que representa un número racional $\frac{p}{q}$. Sus variables miembro son de tipo entero, llamados `p` y `q`.

    b) Defina el constructor de la clase `Fraccion`, que tiene como parámetros enteros `p` y `q`. Por defecto, toman los valores 0 y 1, respectivamente.

    c) Defina la **diferencia** de números racionales mediante una función miembro `Diferencia` de la clase `Fraccion`, que realiza la operación:

    $$\frac{p_1}{q_1} - \frac{p_2}{q_2} = \frac{p_1 q_2 - p_2 q_1}{q_1 q_2}$$

    Esta función toma un parámetro de entrada de tipo `Fraccion` y devuelve un objeto de tipo `Fraccion`.

    d) Defina el **producto** de números racionales mediante una función miembro `Producto` de la clase `Fraccion`, que realiza la operación:

    $$\frac{p_1}{q_1} \cdot \frac{p_2}{q_2} = \frac{p_1 p_2}{q_1 q_2}$$

    Esta función toma un parámetro de entrada de tipo `Fraccion` y devuelve un objeto de tipo `Fraccion`.

    e) Defina el **cociente** de números racionales mediante una función miembro `Cociente` de la clase `Fraccion`, que realiza la operación:

    $$\frac{\frac{p_1}{q_1}}{\frac{p_2}{q_2}} = \frac{p_1 q_2}{q_1 p_2}$$

    Esta función toma un parámetro de entrada de tipo `Fraccion` y devuelve un objeto de tipo `Fraccion`.

    f) Defina un método `aTexto` de la clase `Fraccion`, que devuelve:

    - "0" si $p = 0$
    - "1" si $p = q$
    - "n/m" si es una fracción irreducible (es decir, después de eliminar los factores comunes de p y q)

    g) Defina una clase `Matriz`, que representa una matriz $n \times n$ de números racionales. Tiene una variable entera `n` y un puntero doble `Mat` de tipo `Fraccion`.

    h) Defina el constructor de la clase `Matriz`, que recibe como parámetro un entero `n`, y asigna memoria dinámica al puntero doble `Mat`.

    i) Defina un método llamado `EliminacionGauss`, que mediante operaciones elementales lleva la matriz `Mat` a su forma escalonada.

Ejemplo: Si el ingreso de las entradas de la matriz se realiza mediante:

```
int n = 3;
Fraccion entradas[] = { {1, 1}, {1, 2}, {1, 3},
                        {1, 1}, {5, 6}, {7, 12},
                        {1, 2}, {1, 4}, {11, 30} };
```

Entonces el programa debe mostrar la salida correspondiente en forma escalonada.