UNIVERSIDAD MAYOR DE SAN ANDRES
FACULTAD DE TECNOLOGÍA
INFORMATICA SUPERORI I
GESTIÓN I/2025

PRACTICA N°1

PARTE I: ESTRUCTURAS CONDICIONALES

1. El promedio de prácticas de un curso se calcula en base a cuatro prácticas calificadas, de las cuales se elimina la nota menor y se promedian las tres notas más altas. Elabore un algoritmo que imprima: la nota eliminada y el promedio de prácticas de un estudiante.
   Ejemplo:
   Entrada → 9 8 7 10 **Salida** → Nota eliminada: 7 Promedio: 9

2. Verificar si la fecha proporcionada es correcta. La entrada consiste de tres números enteros a, b, c. Representando el día, mes y año.
   Ejemplo:

Entrada:	**Salida:**
29 02 2003	Fecha Incorrecta
30 07 2003	Fecha Correcta
05 13 1991	Fecha Incorrecta

3. Escriba un programa que tome como entrada un número M donde $1 \leq M \leq 12$ y devuelva el nombre del mes correspondiente en español.
   Ejemplo:

Entrada:	**Salida:**
1	Enero
11	Noviembre

4. Realizar un programa para un conjunto de n datos reales se desea determinar el mayor de los datos negativos y cuantas veces aparece. Por ejemplo:
   Para n=6, datos: 4,-5,6,-2,5,-3.
5. Recibir N número enteros, entre los cuales hay positivos, negativos y ceros. Se debe calcular el promedio de los que fueran simultáneamente positivo y múltiplos de 5. Si no hubiera ningún número con estés condiciones, mostrar un mensaje "inexistente".

PARTE II: ESTRUCTURAS REPETITIVAS

1. Elabore un algoritmo que dado un número N, halle la suma de los números pares e impares, muestre por pantalla el resultado de ambas sumas.
   Ejemplo:

Entrada:	**Salida:**
10	30
	25

2. Escribir un programa para mostrar por pantalla la siguiente pirámide de dígitos para un valor de n entre 1 y 9 (validarlo). Por ejemplo, para n=5:

```
12345
1234
1 2 3
12
1
```

3. Dado un número natural, generar:
   Ejemplo, para n=5:

```
12345
23451
34512
45123
51234
```

4. Dado un número natural, generar:
   Ejemplo, para n=5:

```
    1
   121
  12321
 1234321
123454321
    5
    5
    5
```

5. Realizar un programa que realice el siguiente datagrama con un N introducido por teclado, por ejemplo:
   Para un N=5

```
0  0  0  0  1
0        1
      1
   1        0
1  0  0  0  0
```

PARTE III: LISTAS Y DICCIONARIOS

1. Dada una cadena se pide eliminar las vocales y duplicar las consonantes.
   Ejemplo:

Entrada:	**Salida:**
caminante no hay camino	ccmmnnnntt nn hhyy ccmmnn
hola mundo	hhll mmnndd

2. Dada una lista de palabras, contar cuántas vocales hay en total.
3. Dados dos listas: nombres y edades, combinar ambas y encontrar el tercer nombre con la menor edad.
4. Convertir 'a' → 1, 'e' → 2, 'i' → 3, 'o' → 4, 'u' → 5 en cada palabra de una lista.
5. Dadas dos listas de números, sumarlas, mostrar la lista resultante y de esta misma obtener los múltiplos de 3
6. Dado un diccionario, eliminar las claves que comiencen con una vocal.

7. Realizar un programa que tome una cadena de texto y devuelva un diccionario donde las llaves sean las palabras y los valores la cantidad de veces que aparecen.
   Entrada: "python es genial y python es poderoso"
   Salida: {'python': 2, 'es': 2, 'genial': 1, 'y': 1, 'poderoso': 1}
8. Si el valor es un número par, cambiarlo a 'Par'. Si es impar, cambiarlo a 'Impar'.
   Entrada: lista_numeros = [3, 8, 15, 22, 7, 10, 5]
   Salida: ['Impar', 'Par', 'Impar', 'Par', 'Impar', 'Par', 'Impar']
9. Dado un diccionario con palabras como valores, contar la frecuencia de cada vocal.
10. Dados dos diccionarios, combinarlos y ordenar sus claves.

PARTE IV: POO

1. Usando programación orientada a objetos Construya una clase para efectuar las siguientes operaciones aritméticas de dos números ingresados diferentes de cero.

OPERACIONES
primer valor segundo valor
Potencia () Raíz Cuadrada () Multiplicación () División ()

2. Crear la clase CuentaBancaria para que tenga un método transferir que permita transferir dinero entre dos cuentas.

CUENTA BANCARIA
Primera cuenta Saldo cuenta Segunda cuenta Saldo cuenta
Transferencia entre cuentas

   NOTA: Si se desea, se puede modificar en caso de querer aplicar herencia a este ejercicio.

3. Crea una clase Coche con los atributos privados _marca, _modelo y _velocidad. Implementa un método acelerar() que aumente la velocidad y frenar() que la disminuya. Luego, instancia un objeto de esta clase y realiza operaciones con él.
   Ejemplo:

```
mi_coche = Coche("Toyota", "Corolla")
mi_coche.acelerar()
mi_coche.frenar()
print(mi_coche.obtener_velocidad())  # 0
```

4. Crea una clase base CuentaBancaria con atributos privados _titular, _saldo y métodos para depositar y retirar dinero. Luego, crea una subclase CuentaAhorro que tenga un

atributo adicional _tasa_interes y un método aplicar_interes() para aumentar el saldo según la tasa de interés.

   a. Aplicar herencia y encapsulamiento. Donde lo heredado de la clase Padre debe ser: titular y saldo.

5. Crea una clase Animal con un método hacer_sonido(). Luego, crea 4 subclases Perro Gato, Pato y Vaca que sobreescriban este método con sus respectivos sonidos ("Guau", "Miau", etc.). Además, implementa sobrecarga de métodos en Perro para que hacer_sonido() pueda recibir un número de repeticiones del sonido.

PARTE V: TEORÍA

1. La Programación Orientada a Objetos (POO) es un paradigma de programación que organiza el código en _____ y _____ para representar entidades del mundo real y sus interacciones.
2. Una clase en Python se define utilizando la palabra clave _____.
3. ¿Qué es instanciar? Incluya un ejemplo de instanciación
4. El método __init__ en Python se utiliza para _____ cuando se crea una nueva instancia de una clase.
5. Los atributos en una clase son _____ que representan las características o propiedades de un objeto.

Preguntas de verdadero o falso

6. V o F: Un objeto es una instancia de una clase, y puede tener atributos y métodos que le permitan realizar ciertas tareas.
7. V o F: El polimorfismo permite que diferentes clases utilicen el mismo nombre de método, pero con implementaciones diferentes dependiendo de la clase en la que se encuentre.
8. V o F: La herencia en POO permite que una clase derive de otra clase, heredando sus atributos y métodos sin necesidad de reescribir el código.

Preguntas de desarrollo

9. Explica qué es el encapsulamiento y cómo se logra en Python. Da un ejemplo donde se utilicen métodos getter y setter para controlar el acceso a un atributo.
10. Explique que es herencia
11. Explique que es polimorfismo en Python
12. ¿Qué es la abstracción en la Programación Orientada a Objetos?