TRABAJO PRÁCTICO N°1

Asignatura: Programación Orientada a Objetos

Ejercicio 1

Indicar a qué tipo de dato en Python corresponden los siguientes valores

a) 'Hola' b) 7.35 c) 0 d) 10 e) True f) '123' g)'CP8000'

Ejercicio 2

Indicar si las siguientes expresiones son válidas en Python.

a) 10.25+7.02 b) 10.25+3 c) 10.25+'3' d) '10.25'+'3' e) '10.25+3'

Ejercicio 3

Evaluar, si es posible, las siguientes expresiones lógicas. En los casos que no sea posible, explicar el motivo. Responder considerando que las variables indicadas tienen los siguientes valores: a = 5, b = 8, c = 1.5, d= True , e = '10'.

a) a < 8 b) a < b c) a <= b d) (a-b) < c e) c < e f) not(d) g) e > '4'

Ejercicio 4

El operador de módulo (**%**) permite obtener el resto de la división entera, y es útil en muchas aplicaciones. En cada uno de los siguientes incisos, indicar cómo puede emplearse para resolver el problema planteado:

a) Determinar si un número es par o impar.
b) Determinar si un valor ingresado por el usuario es múltiplo de 4.
c) Determinar si un valor ingresado por el usuario es múltiplo de 3 ó divisor de 36.
d) Determinar si un valor ingresado por el usuario es múltiplo de 3 y divisor de 36.

Ejercicio 5

Implementar un programa que permita generar cuatro números aleatorios y calcule el promedio.

Ejercicio 6

El lenguaje Pyhton provee diversas formas de utilizar cadenas para mostrar resultados en la pantalla. Considerando que se tienen las variables marca = "Motorola", modelo = "G8", memoria = "3", indicar tres o más formas distintas de generar la frase "Su celular es el modelo G8 de la marca Motorola y tiene 3GB de RAM".

Ejercicio 7

En una aplicación para generar una cuenta de usuario, se aplica una serie de controles para que el usuario ingrese su nombre de usuario de manera correcta, teniendo en cuenta que debe respetar las siguientes condiciones:

a) Debe contener al menos 6 caracteres.
b) Debe comenzar con una letra mayúscula y todas las demás en minúsculas.
c) No debe tener espacios al comienzo ni al final.
d) No puede contener números.

Indicar las expresiones necesarias para realizar cada uno de los controles sobre la variable nombreUsuario. Luego implementar un programa que permita al usuario ingresar el nombre deseado e indicar si es correcto o no cumple alguna de las condiciones solicitadas.
Agregar al menos dos condiciones más (a elección) e implementarlas junto a las anteriores.

Ejercicio 8

Implementar un programa que permita al usuario ingresar tres valores: valorInicial, valorFinal y salto. En base a esos tres valores, mostrar la lista de todos los números a partir de valorInicial de salto en salto hasta valorFinal. Resolver el ejercicio utilizando la estructura de control for y luego la estructura de control while.
Ejemplo: si valorInicial = 10, valorFinal = 30 y salto = 5, se deberán mostrar los números "a partir de 10, de 5 en 5 hasta llegar a 20", es decir: 10, 15, 20, 25, 30.

Ejercicio 9

Implementar un programa que permita ingresar valores enteros positivos. El ingreso termina cuando el usuario no desee ingresar más valores, para lo cual se deberá definir un criterio que indique esta decisión. Luego deberá calcular y mostrar el promedio, la cantidad de valores ingresados, el mayor y menor valor.

Ejercicio 10

a) Crear una lista que contenga valores enteros desde 3 a 12 incluidos. Utilizar el identificador listaNumeros.
b) Indicar el resultado de las siguientes operaciones realizadas sobre la lista del inciso anterior:

i) x = len(listaNumeros)
ii) x = listaNumeros[3]
iii) x = listaNumeros[:3]
iv) x = listaNumeros[3:]
v) x = listaNumeros[3:6]
vi) x = listaNumeros[1:8:2]
vii) x = listaNumeros[-1]
viii) x = listaNumeros[-6:-2]
ix) x = listaNumeros[: : -1]

Ejercicio 11

Considerar que se tiene una lista1 = [7, 4, 5, 3], indicar el resultado de realizar las siguientes operaciones en forma consecutiva (es decir, la instrucción del inciso ii se aplica sobre el resultado del inciso i y así sucesivamente).
i) lista1.append(6)
ii) lista1[2] = 10
iii) lista1.insert(3, 8)
iv) lista1.remove(4)
v) lista1.pop(2)
vi) lista1.extend([1, 2])
vii) lista1.reverse()
viii) lista1.sort()

ix) lista1.reverse()
x) lista1.clear()
xi) lista1.pop()

Ejercicio 12

a) Implementar una función que permita extraer la última cifra de un número entero positivo (es decir, la cifra de unidad).
b) Implementar una función que permita extraer todas las cifras excepto la última cifra de un número entero positivo (es decir, todas menos la cifra de unidad).
c) Implementar una función que devuelva la cantidad de cifras que contiene un número entero positivo.

Ejercicio 13

Implementar una función contieneCifra(numero, cifra), que devuelva un valor booleano indicando si la cifra aparece en el número dado.
Ejemplos: b = contieneCifra(123, 2) # True b = contieneCifra(123, 5) # False

Ejercicio 14

a) Implementar una función de devuelva el mayor de dos números reales.
b) Utilizando la solución del inciso anterior, implementar una función que devuelva el mayor de tres números reales.

Ejercicio 15

Implementar las funciones necesarias para realizar las siguientes operaciones de un software que permite trabajar con polinomios cuadráticos.
a. A partir de los coeficientes de la forma polinómica, implementar una función que genere una cadena de caracteres con la expresión correspondiente.
*Ejemplo: si a=2,b=4,c=5, generar la cadena "P(x)=2*x^2 + 4*x + 5".*
b. A partir de los coeficientes de la forma polinómica, determinar si tiene raíces reales y en caso afirmativo calcularlas.
c. Usando el subprograma del inciso anterior, generar una cadena de caracteres con la forma factorizada.
Ejemplo: si a=6 y las raíces son x1=2, x2=5, generar la cadena "P(x)=6(x-2)(x-5)"
d. Implementar una función que permita evaluar el polinomio en un valor de x dado.
d. Utilizando la función del inciso anterior, implementar un subprograma para determinar el punto x de un intervalo dado donde el polinomio toma el menor valor. Los extremos del intervalo [xmin , xmax] deben ser recibidos por parámetro, además de un parámetro adicional maxdif , que indicará la distancia entre dos puntos de evaluación consecutivos.
Ejemplos:
Si el intervalo es [1,4] y maxdif es 0.1, el polinomio se deberá evaluar en los puntos 1; 1.1; 1.2; 1.3; ...; 3.8; 3.9; 4. En cambio, si maxdif es 1, el polinomio se deberá evaluar en los puntos 1; 2; 3; 4.

Ejercicio 16 – Resumen del apunte

*Nota: responder las siguientes preguntas en base a la lectura previa del apunte teórico de la **Unidad 1 - Primera parte**.*

a. Indicar los tres aspectos que podemos considerar para describir la evolución de los lenguajes de programación.
b. ¿Cuál es la principal función de un lenguaje ensamblador?
c. De acuerdo al tipo de instrucciones, indicar cuáles son los tres tipos de lenguajes que podemos mencionar.
d. Según la respuesta anterior, ¿a qué grupo pertenece el lenguaje Python?
e. Indicar la clasificación de lenguajes según su forma de ejecución y describir la principal diferencia entre ambos.
f. Indicar la clasificación de lenguajes según su conceptualización.
g. Describir el concepto de paradigma de programación.
h. Mencionar los diferentes paradigmas de programación.
i. Indicar cuáles son los principales problemas que se intentan abordar mediante la Programación Orientada a Objetos.

AYUDAS:

Para generar valores aleatorios, será necesario importar el módulo ***random***. Así se podrán utilizar las funciones ***randint(min,max)*** y ***random()***.

La función **divmod**(dividendo, divisor) devuelve el cociente y resto de una división entera.

Ejemplo: div = divmod(14, 5) # retorna (2, 4), es decir cociente 2 y resto 4

Algunos métodos que pueden ser de utilidad en el manejo de cadenas:

len(cadena): devuelve la cantidad de caracteres que contiene la variable cadena.
cadena.***capitalize***(): convierte a mayúsculas el primer caracter de la letra de cadena, todos los demás en minúscula.
cadena.***strip***(): elimina espacios en blanco al comienzo y al final de cadena.
cadena.***endswith***(valor): devuelve True si la cadena termina con el valor indicado.
cadena.***startswith***(valor): devuelve True si la cadena empieza con el valor indicado.
cadena.***upper***(): convierte la cadena a mayúsculas.
cadena.***lower***(): convierte la cadena a minúsculas.
cadena.***isalpha***(): devuelve True si todos los caracteres de cadena están en el alfabeto.
cadena.***isnumeric***(): devuelve True si todos los caracteres de cadena son números (0-9).