**Evaluación final Programación**

3.La sintaxis:  
  
tup = [10, ‘Chelsea’, 20], ¿define específicamente una?

a)   Tupla

b)   Lista

c)   Diccionario

d)   Secuencia

e)   Conjunto

4.Para acceder a los atributos y métodos de una clase principal, dentro de una clase secundaria en Python se utiliza:







Super



5.¿Por qué la expresión en Python, def add(\*args), es incorrecta? 

a)   No es incorrecta, funciona bien

b)   El compilador de Python no interpreta el apuntador

c)   Le hace falta el bloque de código o la palabra pass

d)   Una función suma (add) necesita de más de una variable

e)    Le faltan las llaves del bloque de código

6.Cuando se habla de Posicionales, Palabras Clave, Arbitrarios y Predeterminados en Python se hace referencia a:

a)  Tipos de datos

b)   Argumentos

c)   Parámetros

d)   Variables

e)   Funciones

7.Una buena definición de qué son los parámetros sería:

a)   Son los valores que se le pasan a una función al invocarla

b)    Son los nombres enumerados en la definición de una función

c)    Son los valores definidos dentro de la función

d)    Son los nombres de definición de la función

e)    Son los argumentos de la función

8.Una buena definición de qué son los argumentos sería:

a)    Son los valores que se le pasan a una función al invocarla

b)    Son los valores que se le pasan a una función al definirla

c)    Son los valores definidos dentro de la función

d)    Son los nombres de definición de la función

e)    Son los argumentos de la función

9.Para que la expresión  
  
for i in range(5):  
if i == 3:  
xxxx  
print(i)  
  
Dé como resultado:  
  
0  
1  
2  
4  
  
Es necesario reemplazar las xxxx por:

a)    break

b)    return

c)    next

d)    continue

e)    jump

10.Para que la expresión  
  
for i in range(5):  
         if i == 3:  
         xxxx  
         print(i)  
  
Dé como resultado:  
  
0  
1  
2  
  
Es necesario reemplazar las xxxx por:

a)    continue

b)    return

c)    next

d)    break

e)    jump

11.Para obtener como resultado   
  
a = 10, b = [11, 12]  
  
De la expresión  
  
items = [10, 11, 12]  
  
Se debe hacer:

a)    a, b = items

b)    a = 10 , b = [11, 12]

c)    a, \*\*b = items

d)    a, \*b = items

e)    a = items[0], b  = items[1:2]

12.La segmentación en Python acepta los siguientes tipos de datos:

a)    lista, tupla, cadena

b)    enteros, cadena, flotantes

c)    matrices, flotantes, cadena

d)    Complejos, flotantes, enteros

e)    matrices, listas, complejos

13.La siguiente expresión en Python  
  
a = {1: 2, 2: 3}  
a.pop(2)  
a  
Da como resultado al ejecutar la segunda y tercera línea:

a)    1 {1: 2}

b)    1 {3}

c)    3 {1: 2}

d)    3 {2: 3}

e)    3 {2}

14.Utilizando paréntesis para agrupar elementos, sin cambiar el orden, ¿cuál es la combinación correcta para que la siguiente expresión dé como resultado 17?  
  
x= 1 / 2 + 3 // 3 + 4 \*\* 2  
  
print(x)

a)    x = 17  print(x)

b)    x = 1 + (4)\*\*2  print(x)

c)    x = (1 / 2 + 3) // (3 + 4 \*\* 2)  print(x)

d)    x = (1/2+3) // 3+4\*\*2  print(x)

e)    x = 1/2+3//(3+4)\*\*2  print(x)

15.  
La expresión **print(1 / 2 + 3 // 3 + 4 \*\* 2)"**  en Python, devuelve como resultado:

a)    17.5

b)    17

c)    3.5

d)    8

e)    8.5

16.El resultado de la expresión  **print(2 \*\* 2 \*\* 3)**  en Python, da como resultado:

a)    64

b)    12

c)    256

d)    128

e)    Syntax error, print no se usa como calculadora

17.La expresión  **print(-6 // 4)**da como resultado en Python

a)    -2

b)    2.0

c)    -1.5

d)    1.5

e)    Syntax error, print no se usa como calculadora

18.La división en Python se expresa utilizando el símbolo ""/"". ¿Cuál sería el resultado de la división **print(6 / 3)** en Python?

a)    2

b)    2.0

c)    6/3

d)    "2"

e)    Syntax error, print no se usa como calculadora

19.Para este caso se va a suponer que la velocidad de la luz es aproximadamente 300.000.000 m/s (Trescientos millones de metros por segundo), un valor más aproximado es 299.792.458 pero a modo de ejemplo se utilizará el valor inicial, ¿cómo se podría expresar la velocidad de la luz utilizando notación científica en Python?

a)    3\*10^8

b)    3e10^8

c)    3E8

d)    pow(3, 8)

e)    3\*\*8

20.Al ejecutar el código **class Perro: pass,** se crea:

a)    Una función vacía que puede ser utilizada como una clase si se complementa.

b)    Una clase vacía

c)    Una clase con error ya que necesita ser inicializada

d)    Un objeto de la clase Perro

21.La expresión   **1 // 2** da como resultado

a)    0

b)    1/2

c)    SyntaxError: invalid syntax

d)    IndentationError: expected an indented block

e)    0.5

22.¿Cuál de las siguientes afirmaciones es correcta a la hora de hablar de Python?

a)    Python es un lenguaje interpretado, contrario a otros lenguajes como C, C++ o C#, Python no necesita de compilados antes de ser ejecutado. Hay otros lenguajes interpretados como PHP y Ruby.

b)    Python es un lenguaje compilado, contrario a otros lenguajes como PHP o Ruby, Python necesita ser compilado antes de ser ejecutado. Hay otros lenguajes compilados como Java.

c)    Python es un lenguaje especial, contrario a otros lenguajes que son Compilados o Interpretados, Python es inicialmente compilado para luego ser interpretado para ser ejecutado en cualquier sistema operativo.

d)    Python es un lenguaje de ejecución en tiempo real, no necesita ser interpretado ni compilado ya que cuenta con su propio motor de ejecución en tiempo real.

Tercera parte Prueba Programación

23.Las siguientes son características de Python:

a)    Compilado, fuertemente tipado, incluye extensiones basadas en MatLab, propósito especializado

b)    Interpretado, dinámico, orientado a objetos, incluye extensiones basadas en C, propósito generalizado

c)    Compilado, debilmente tipado, incluye extensiones basadas en C, propósito general

d)    Compilado, dinámico, orientado a objetos, incluye extensiones basadas en C, propósito generalizado

24.El sistema que utiliza Python para asegurarse que los nombres son únicos, ¿es conocido como?

a)    namespacing

b)    nameboard

c)    space of names

d)    namepath

e)    namespace

25.El código  
  
a=2               
def add():  
         b=3                         
         c=a+b  
         print(c)  
  
devuelve:

a)    5

b)    IndentationError: expected an indented block

c)    Nada

d)    0

e)    SyntaxError: invalid syntax

26.Gracias a la elevación o hoisting en Python, ¿son los bloques de código 1 y 2 equivalentes?:  
  
1.  
repetir\_funciones()  
  
def imprime\_Cosas():  
         print("La clase esta genial")     
         print('Python es lo maximo')  
  
def repetir\_funciones():  
         imprime\_Cosas()  
         imprime\_Cosas()  
  
2.  
def repetir\_funciones():  
         imprime\_Cosas()  
         imprime\_Cosas()  
  
def imprime\_Cosas():  
         print("La clase esta genial")     
         print('Python es lo maximo')  
  
repetir\_funciones()

a)    Si, pero (1) retrasa el flujo de ejecución

b)    No

c)    Si

d)    No, pero se obtiene el mismo resultado

27.Las estructuras de par llave: valor en Python son conocidas como:

a)    Diccionarios

b)    Tuplas

c)    Listas

d)    Conjuntos

e)    Key:Value

f)      Range

28.La conversión de tipos en Python es muy utilizada para operar entre diferentes tipos de datos, ¿cuáles de las siguientes conversiones, (Type Conversion), son válidas?

a)    int(), float(), imag(), list(), hex(), oct()

b)    int(), complex(), list(), hex(), oct(), set()

c)    int(), float(), real(), list(), hex(), oct()

d)    int(), float(), list(), hex(), tuple(), map()

e)    int(), float(), ord(), list(), oct(), conj()

29.¿Qué es \_\_init\_\_?

a)    Es la forma de inicializar una variable

b)    Se utiliza para inicializar funciones

c)    Es un método o constructor

d)    Es un método que algunas clases poseen

30.¿Qué es self?

a)    Es un parámetro opcional de una clase

b)    Es una instancia de una función

c)    Es la referencia al método init del objeto recien creado

d)    Es un objeto o instancia de una clase

e)    Es un llamado a una función anónima

31.El código:  
  
def mi\_funcion():  
  a=int(input("ingresa un numero: "))  
  if a>2:  
         for x in range(2,a):  
         if(a%x)==0:  
         print("no")  
         return False  
      
         print("si")  
         return True  
  else:  
         print("no me sirve")  
mi\_funcion()  
  
identifica si un número es:

a)    Real

b)    Imaginario

c)    Primo

d)    Par

e)    Impar

32.El bloque 'else' del try-except-else se ejecuta cuando:

a)    Siempre que ingrese al try

b)    Cuando no se produce ninguna excepción

c)    Cuando se produce alguna excepción

d)    Cuando se produce una excepción en el bloque excepto

e)    Nunca

FIN