



Universidad del Valle

Facultad de Ingeniería
Escuela de Ingeniería de Sistemas y Computación
Fundamentos de programación imperativa

NORMAS PARA LA ENTREGA DE LOS LABORATORIOS

- Coloque el nombre de los integrantes del grupo (máximo cuatro estudiantes), nombre del profesor, número del grupo de Algoritmia y número de laboratorio presentado en el encabezado de todos sus programas.
- **Lugar y Medio de Entrega:** Todos los archivos que se soliciten en el informe se deben subir al campus virtual en el respectivo enlace del laboratorio.
- Debe entregar el laboratorio por el campus virtual y realizar una sola entrega por grupo. En caso de recibir más de una entrega por grupo sólo se calificara una de ellas, así presenten diferencias entre ellas.
- La fecha máxima de entrega será el 10 de Abril de 2022 a las 23:59:59 hora de Colombia por el campus virtual.
- Se permiten entregas atrasadas, sin embargo, estas tendrán una sanción de 0.3 por hora o fracción. Es decir, si entrega el 11 de Abril a las 00:00:01 tendrá 0.3 de sanción, si entrega a las 01:00:01 tendrá 0.6 y así sucesivamente.
- **Plazo:** Los estudiantes deben subir los archivos antes del día y la hora establecida por el profesor en el campus virtual, el vínculo para esta actividad se deshabilitará automáticamente una vez se cumpla el plazo.
- Durante el curso no se recibirán informes de laboratorio enviados por correo electrónico y tampoco se podrá entregar informes al monitor.

Laboratorio No. 2 Estructuras de Decisión y Funciones Fecha de entrega: abril 01 de 2022

Objetivos:

1. Identificar operadores relacionales y lógicos.
2. Construir y aplicar expresiones condicionales.
3. Familiarizar al estudiante con las distintas estructuras condicionales en Python.
4. Familiarizar al estudiante con las funciones en Python.
5. Comprender el uso y la sintaxis de una función.

Para cada ejercicio los entregables son:

1. Análisis del problema
2. Algoritmo en pseudocódigo
3. Prueba de escritorio.....
4. Algoritmo en Python.....
5. Evidencias de ejecución del programa.....

Se debe entregar un informe PDF con los puntos 1, 2, 3, y 5 de la metodología. El punto 4 corresponde a los archivos **.py**

Problema 1. *Calculando el valor de compra de unas computadoras*

Una fábrica de computadoras planea ofrecer a los clientes un descuento que dependerá del número de computadoras que compre. Si las computadoras son menos de 5 se le dará un 10% de descuento sobre el total de la compra, si el número de computadoras es mayor o igual a 5 pero menos de 10, se le otorgará un descuento del 20%; y si son 10 o más se le dará un 40% de descuento. El precio de cada computadora deberá ser ingresado.

Realice un programa que solicite los datos para la venta: Nombre de Cliente, Cantidad de Computadoras a comprar y precio unitario. El programa deberá tener una función que calcule e imprima: Precio total de las computadoras, Descuento Obtenido y Valor total a pagar.

La función deberá tener la siguiente firma:

```
#Función calculoVenta  
def calculoVenta(nombre,cantidad, precio)
```

Asuma que el programa deberá registrar los datos para 3 ventas, tal como se ve en el siguiente ejemplo:

```

Venta No. 1
=====
Nombre del Cliente : Pedro
Cantidad de computadores a comprar : 10
Precio Unitario por Computadora: 1200000
=====
Precio de las computadoras : 12000000.0 pesos
Descuento obtenido ..... : 4800000.0 pesos
Valor total a pagar es ... : 7200000.0 pesos
=====

Venta No. 2
=====
Nombre del Cliente : Maria
Cantidad de computadores a comprar : 3
Precio Unitario por Computadora: 1200000
=====
Precio de las computadoras : 3600000.0 pesos
Descuento obtenido ..... : 360000.0 pesos
Valor total a pagar es ... : 3240000.0 pesos
=====

Venta No. 3
=====
Nombre del Cliente : Juan
Cantidad de computadores a comprar : 7
Precio Unitario por Computadora: 1200000
=====
Precio de las computadoras : 8400000.0 pesos
Descuento obtenido ..... : 1680000.0 pesos
Valor total a pagar es ... : 6720000.0 pesos
=====

```

Problema 2 . Calculando la nota definitiva de un estudiante

Un estudiante está cursando tres asignaturas y desea saber cuál será su nota final en una de las asignaturas, dado el nombre de la asignatura, la nota del examen parcial y las notas de tres tareas. Se debe tener en cuenta que:

- La calificación de Matemáticas se obtiene de la siguiente manera: Examen 90% y promedio de las tareas 10%.
- La calificación de Física se obtiene de la siguiente manera: Examen 80% y promedio de las tareas 20%.
- La calificación de Química se obtiene de la siguiente manera: Examen 85% y promedio de las tareas 15%.

Realice un programa que contenga una función que reciba el nombre de la asignatura y solicite la nota del examen, la nota de la tarea 1, la nota de la tarea 2 y la nota de la tarea 3, calcule la nota definitiva de la materia e imprima si ganó o perdió la materia. Una

asignatura se gana con una nota final igual o superior a 3.0. El programa debe utilizar la función 3 veces, una vez para cada una de las materias.

La función deberá tener la siguiente firma:

```
#Función nota de una materia
def notaFinal(nombreMateria)
```

A continuación, se presenta un ejemplo a ejecutar:

```
Ingrese nombre de la materia : Fisica
=====
Ingreso de notas para : Fisica
Ingrese la nota del parcial : 5
Ingrese la nota de la tarea 1 :4
Ingrese la nota de la tarea 2 :3
Ingrese la nota de la tarea 3 :4
La nota definitiva en Fisica es: 4.733333333333333
Ganó la materia
=====

Ingrese nombre de la materia : Matematicas
=====
Ingreso de notas para : Matematicas
Ingrese la nota del parcial : 4
Ingrese la nota de la tarea 1 :3
Ingrese la nota de la tarea 2 :5
Ingrese la nota de la tarea 3 :3
La nota definitiva en Matematicas es: 3.966666666666667
Ganó la materia
=====

Ingrese nombre de la materia : Quimica
=====
Ingreso de notas para : Quimica
Ingrese la nota del parcial : 2
Ingrese la nota de la tarea 1 :3
Ingrese la nota de la tarea 2 :1
Ingrese la nota de la tarea 3 :2
La nota definitiva en Quimica es: 2.0
Perdió la materia
=====
```

Problema 3

El Departamento de Salud Ocupacional de la Universidad del Valle, en su campaña **Salud para Todos**, requiere de un programa que le permita determinar el Índice de Masa Corporal (IMC) y el estado de obesidad de su comunidad educativa: estudiantes, profesores y empleados.

Estado	IMC
Bajo Peso	< 18.5
Peso normal	18.5 – 24.9
Sobrepeso	25.0 – 29.9
Obesidad Tipo I	>= 30.0

El IMC = $\text{Peso en Kilos} / (\text{estatura en Mts})^2$

El programa deberá solicitar el nombre de la persona, el peso y la estatura, como se muestra a continuación:

```
>>>
Nombre .... : Pedro
Peso ..... : 45
Estatura .. : 1.8
```

Una vez capturados los datos se deberá mostrar los siguientes valores:

```
=====
Nombre ..... : Pedro
Peso ..... : 45.0
Estatura ..... : 1.8
IMC ..... : 13.888888888888888
Estado Obesidad : Bajo Peso
>>> |
```

El programa deberá tener dos funciones, una recibirá el peso y la estatura de la persona, con estos datos calculará el IMC y deberá regresar dicho valor. La otra función deberá recibir el valor del IMC y retornará el estado de Obesidad.

La función para el cálculo del IMC deberá tener la siguiente firma:

```
#Función que calcula el IMC
def determinarIMC(peso, estatura)
```

La función para determinar estado de obesidad deberá tener la siguiente firma:

```
#Función que indica el estado de obseidad
def determinarObesidad(imc)
```

Rubricas de evaluación

Cumplimiento de las reglas del taller

Concepto	Nivel 0 (0 puntos)	Nivel 1 (5 puntos)	Nivel 2 (10 puntos)
Organización de la entrega	No realiza la entrega	Entrega más de un archivo .py por cada punto del taller o sus nombres no son claros con respecto al punto que resuelven.	Entrega exactamente un archivo .py para cada punto del taller y sus nombres son claros, por ejemplo: punto1.py, punto2.py, etc.
Formato del informe	No entrega el informe	Entrega el informe en un formato diferente al formato PDF	Entrega el informe en formato PDF
Archivo de entrega	No realiza la entrega	Entrega un archivo comprimido que contiene archivos comprimidos dentro	Entrega un archivo comprimido que únicamente contiene el informe en formato PDF y los archivos .py de los puntos
Nombres de los estudiantes dentro del código	No incluye dentro de las primeras líneas de los archivos un comentario que tenga los nombres y códigos de los estudiantes del grupo de trabajo	Incluye dentro de las primeras líneas de los archivos un comentario que tenga los nombres y códigos de los estudiantes del grupo de trabajo, pero hace falta en uno o más de los archivos .py o bien a alguno de los estudiantes le hace falta su código estudiantil	Incluye dentro de las primeras líneas de los archivos un comentario que tenga los nombres y códigos de los estudiantes del grupo de trabajo en todos los archivos .py

Estas rubricas corresponden a 40 puntos del taller.

Cada uno de los puntos

Estas rubricas corresponden a la evaluación de cada uno de los puntos de taller de forma independiente de acuerdo a los logros del curso.

Logro	Nivel 0 0 puntos	Nivel 1 10 puntos	Nivel 2 20 puntos
Implementa en un lenguaje de programación un algoritmo que da solución a un problema que incluya entrada y salida , estructuras de decisión, ciclos, arreglos y/o funciones	No implementa el código en Python	Implementa el código en Python, pero no incluye input para solicitar la información o el código tiene algún error con respecto a la especificación o no muestra la salida con un print.	Implementa el código en Python, incluye input para solicitar la información al usuario y el código no tiene ningún error con respecto a la especificación y muestra la salida con un print.
Implementa en un lenguaje de programación un algoritmo que da solución a un problema que incluya entrada y salida, estructuras de decisión , ciclos, arreglos y/o funciones	No realiza el punto	Usa estructuras de decisión, sin embargo presenta errores lógicos de su diseño que no permite establecer correctamente los casos que presenta el problemas	Usa estructuras de decisión y estas trabajan correctamente el problema.
Implementa en un lenguaje de programación un algoritmo que da solución a un problema que incluya entrada y salida, estructuras de decisión, ciclos, arreglos y/o funciones	No usa funciones para resolver el problema	Implementa las funciones solicitadas, pero estas no tienen valores de retorno y se usan print internamente para mostrar la solución	Implementa las funciones solicitadas y estas tienen los valores de retorno esperados para cada caso.
Rastrea la ejecución de segmentos de código y describe su cómputo	No realiza la prueba de escritorio o no adjunta capturas sobre evidencias de ejecución	Realiza la prueba de escritorio, pero esta presenta fallas con respecto al pseudocódigo propuesto o bien no realiza al menos 3 ejemplos conceptuales de su algoritmo o adjunta capturas de menos de 3 ejemplos de ejecución en el código dentro del informe.	Realiza la prueba de escritorio correctamente con respecto al pseudocódigo propuesto y realiza al menos 3 ejemplos conceptuales de su algoritmo y adjunta capturas de al menos 3 ejemplos de ejecución en el código dentro del informe.

Aplica una metodología de desarrollo para un problema específico y entrega los elementos de cada etapa	No aplica la metodología vista en clase para la solución de problemas en programación o no realiza dos o más pasos de ella o no la evidencia dentro del informe en formato PDF	Aplica la metodología vista en clase, pero aplica incorrectamente uno de sus pasos o no lo realiza. Evidencia la metodología en el informe en formato PDF	Aplica la metodología vista en clase y aplica correctamente sus pasos. Evidencia la metodología en el informe en formato PDF
--	--	---	--

Cada punto tiene 100 puntos, por lo que en total tenemos 300 puntos

Cálculo de la nota del laboratorio

El total del laboratorio **n** es de 320 puntos, por lo que la nota se calculará así tomando en cuenta **p** como los puntos obtenidos en todo el laboratorio.

$$n = 5,0 \frac{p}{320}$$