

Taller de Programación I

Fecha: 12 / 2 / 2019

Cuatrimestre: 2° / 2018

Tema: 2

Padrón: _____ Apellido: _____ Nombres: _____

Ejercicio	1	2	3	4	5	6	7	8	9	10
Resultado										

- 1) ¿Qué es un **functor**? **Ejemplifique**.
- 2) Explique **qué es** cada uno de los siguientes, haciendo referencia a su **inicialización**, su **comportamiento** y el **area de memoria** donde residen:
 - a) Una variable **global static**
 - b) Una variable **local**
 - c) Un **atributo de clase static**.
- 3) **Declare** una clase a elección considerando:
 - **Atributos** que son necesarios
 - **Accesibilidad** a la Clase
 - **Incluir** los operadores **+**, **++**, **>>** (**impresión**), **<<** (**carga desde consola**), **int**
- 4) Escriba un programa que imprima por salida estándar los números entre 1 y 100, en orden ascendente. Se pide que los números sean **contabilizados por una variable global única** y que los **pares sean escritos por un hilo** mientras que los **impares sean escritos por otro**. Contemple la correcta sincronización entre hilos y la liberación de los recursos utilizados.
- 5) ¿Qué es la **compilación condicional**? En qué **etapa del proceso de transformación** de código se resuelve. **Ejemplifique** mediante código C, dando un caso de uso útil.
- 6) Escriba una **función ISO C** que permita **procesar un archivo texto sobre sí mismo**, que contenga **una palabra por línea**. El procesamiento consiste en **ordenar las palabras (líneas) alfabéticamente** considerando que el archivo **no entra en memoria**.
- 7) ¿Por qué las librerías que usan **Templates** se publican con todo el código fuente y no como un .h y .o/.obj?.
- 8) ¿Qué significa la palabra **virtual** antepuesta a un método de una clase? ¿Qué cambios genera a nivel **compilación** y al momento de **ejecución**?
- 9) Escriba un programa (desde la inicialización hasta la liberación de los recursos) que reciba **paquetes de texto delimitados por corchetes angulares ("<...>")** y los imprima **completos por pantalla**. Al recibir un paquete vacío ("**<>**") debe cerrarse ordenadamente. No considere errores.
- 10) **Describe** el concepto de **loop de eventos (events loop)** utilizado en programación orientada a eventos y, en particular, en entornos de interfaz gráfica (GUIs).