**Instituto Superior de Formación Técnica Nº 151 logo151-trans.gif  
Carrera: Analista de Sistemas  
1 Año. Algoritmos y Estructuras de Datos I**

|  |  |
| --- | --- |
| **Trabajo Práctico Nº 10.2** | **Unidad 10.2** |
| **Modalidad:** Semi-Presencial | **Estratégica Didáctica:** Trabajo Individual |
| **Metodología de Desarrollo:** acordar | **Metodología de Corrección:** acordar docente |
| **Carácter de Trabajo:** Obligatorio – con Nota | **Fecha Entrega:** A confirmar por el Docente. |

**Paradigma Orientado a Objetos – Programación Orientada a Objetos**

**Marco Teórico:**

1. Dar un ejemplo de Creación de una clase y de instanciar un Objeto
2. Explicar los alcances de Visibilidad “*public, private o protected”*
3. ¿Qué se entiende por función Miembro de una clase?
4. Qué diferencia hay entre implementar la función miembro dentro y fuera de la clase, dar un ejemplo en código.
5. ¿Qué se entiende por constructor y que es el constructor por defecto?
6. ¿Para qué sirven las listas de Inicialización?
7. ¿Cuándo un Atributo es privado como le asigno valores y como puedo ver su estado?
8. ¿Qué se entiende por Atributo Estático? Dar un ejemplo.
9. ¿Qué se entiende por función miembro Estática? Dar un ejemplo.
10. Que utilidad tiene un Destructor? ¿Qué ocurre si no lo Escribo?
11. Que debería poner dentro de Un Destructor? Dar un ejemplo.
12. Explicar la cita de autor “*La mente humana clasifica los conceptos de acuerdo a dos dimensiones: Variedad y Pertenencia*”
13. Explicar el Concepto de “Generalización y Especialización” en Herencia.
14. Que es el constructor de Copia? ¿Para qué sirve?
15. Describir el concepto de “Jerarquía de Clases”
16. Explicar la diferencia entre “early binding” y “late binding”
17. Dar un ejemplo de código del punto anterior.
18. ¿Que resuelven o permiten las Funciones Virtuales?
19. Qué relación tienen los puntos anteriores con el “Principio de sustitución de Liskov”
20. A qué referencia hace el autor cuando dice “*Existe polimorfismo cuando interactúan la herencia y el enlace dinámico*”.

**Marco Practico:**

**Tener en Cuenta:**  
. Modularizar el Programa.  
. Proteger contra Inclusiones Múltiples.  
. Aplicar Espacios de Nombres.  
. Aplica apropiadamente los conceptos de abstracción, encapsulación y ocultamiento de información.  
. Realiza una apropiada distribución de responsabilidades entre las entidades del espacio de la solución.   
.Desarrolla para reusar.  
.Reusa apropiadamente las entidades desarrolladas en el espacio curricular.  
. Demuestra un uso apropiado de la sintaxis y semántica del lenguaje de programación C++.

**Desarrollar un Programa** que:

1. *Desarrollar una “Solución”* que represente la siguiente estructura jerárquica.
   1. *Que realice la Herencia solicitada*
   2. *Que separe “Interfaz” de “Implementación” en código, modularizándolo.*
   3. *Que permita por menú elegir en tiempo de ejecución el “Tipo” de Objeto.*
   4. *Que permita cargar 5 objetos almacenados en un “Arrays” de figuras pudiendo ser de cualquier tipo derivado.*
   5. *Que Implemente métodos Virtuales la clase “Base”*
   6. *Que permita llamar de forma “Genérica” los métodos sin importar le tipo de objeto del que se trate.*
   7. *Que se Implemente con Punteros del tipo “Figura” apuntando a las clases derivadas*
   8. *Que recorra el Arrays ejecutando el método “Dibujar” usando Polimorfismo (virtual)*

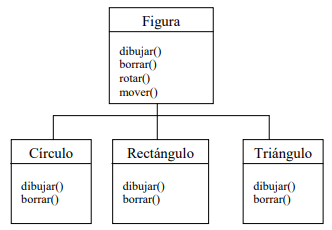


Diagrama de Clase resumido de la solución.

Lic. Oemig José Luis.