**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 10.1** | **Unidad 10** |
| **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. |

**PUNTEROS**

**Marco Teórico:**

1. Explicar con un ejemplo las Direcciones de memoria (tomar un Integer para el Ejemplo)
2. ¿Qué entiende por punteros, que representan?
3. Que utilidades tienen los Operadores Monarios “\*” y “&”?
4. ¿Qué direcciones válidas (valores) tengo para los Punteros?
5. ¿Cuál es el Problema de no inicializar un Puntero?
6. Por qué decimos que es Un valor seguro: NULL?
7. Que entiende por Copia y comparación de punteros
8. Que entendemos por Tipos puntero (piense en tipos de datos)
9. Dar ejemplos de Punteros a estructuras
10. Diferencie con un ejemplo Punteros a constantes y punteros constantes
11. Que relación hay entre Punteros y paso de parámetros (relacione referencia y valor)
12. Explicar la relación entre Punteros y Arrays
13. Explicar el “Esquema de Memoria y datos del programa”
14. Que entiende por Memoria dinámica? ¿Que uso tiene?

**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.** Sin ejecutarlo, ¿qué mostraría el siguiente código?

int x = 5, y = 12, z;

int \*p1, \*p2, \*p3;

p1 = &x;

p2 = &y;

z = \*p1 \* \*p2;

p3 = &z;

(\*p3)++;

p1 = p3;

cout << \*p1 << " " << \*p2 << " " << \*p3;

**2.** ¿Qué problema hay en el siguiente código?

int dato = 5;

int \*p1, p2;

p1 = &dato;

p2 = p1;

cout << \*p2;

**3.** ¿Qué problema hay en el siguiente código?

double d = 5.4, e = 1.2, f = 0.9;

double \*p1, \*p2, \*p3;

p1 = &d;

(\*p1) = (\*p1) + 3;

p2 = &e;

(\*p3) = (\*p1) + (\*p2);

cout << \*p1 << " " << \*p2 << " " << \*p3;

**4.** ¿Cómo declararías un puntero constante p para apuntar a una constante

entera? (Repasa las diapositivas 877‐878 de la presentación del tema.)

**5.** Dado el siguiente tipo:

typedef struct {

string nombre;

double sueldo;

int edad;

} tRegistro;

Y el siguiente subprograma:

void func(tRegistro &reg, double &irpf, int &edad) {

const double TIPO = 0.18;

reg.edad++;

irpf = reg.sueldo \* TIPO;

edad = reg.edad;

}

Reescribe el subprograma para que implemente el paso de parámetros por variable con punteros, en lugar de las referencias que usa ahora (modifica el prorotipo y la implementación convenientemente).

Lic. Oemig José Luis.