CS1102 – PROGRAMACIÓN ORIENTADA A OBJETOS 1
CICLO 2018-2

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

Unidad 6: Punteros
Laboratorio – Sesion 1

http://bit.ly/2p3fgiD

Profesores:

< Juan Flores Moroco >

Telegram:

1. **Configurar tu cuenta**

2. **Link: http://bit.ly/2OW5Ss9**

SISAP:

Evento: SISAP 2018 – 11th International Conference on Similarity Search and Applications
Fechas: October 7-9 Lima, Perú
Resumen: http://www.sisap.org/2018/

The 11th International Conference on Similarity Search and Applications (SISAP) is an annual forum for researchers and application developers in the area of similarity data management. It aims at the technological problems shared by numerous application domains, such as data mining, information retrieval, multimedia, computer vision, pattern recognition, computational biology, geography, biometrics, machine learning, and many others that make use of similarity search as a necessary supporting service.

Inscripciones: https://eventos.spc.org.pe/spire2018/registration_sisap.html

SPIRE:

SPIRE 2018: 25th International Symposium on String Processing and Information Retrieval
Fechas: October 9-11 Lima, Perú
Resumen: https://eventos.spc.org.pe/spire2018/venue.html

SPIRE 2018 is the 25th edition of the annual Symposium on String Processing and Information Retrieval. SPIRE has its origins in the South American Workshop on String Processing, which was first held in Belo Horizonte, Brazil, in 1993. Since 1998 the focus of the workshop has also included information retrieval, due to its increasing relevance to and inter-relationship with string processing.
SPIRE 2018 will be held in UTEC Lima, Peru.
Inscripciones: https://eventos.spc.org.pe/spire2018/registration.html

Logro de la sesión:

Al finalizar la sesión, los alumnos desarrollan sus programas utilizando punteros y arrays dinámicos.

Punteros

Repasar con los alumnos:

El uso del & para declarar una referencia (un alias de la variable)
Jugar con las declaraciones de un puntero, que impriman las direcciones de una variable.
Que quede claro el uso del operador:
& dirección
*** Desreferencia**

Ejemplo 1:

```
#include <iostream>
using namespace std;

int main()
{
  float f=345.87;
  float *pf =&f;

  cout << "valor de f "  << f << "\n";
  cout << "valor de f "  << *pf << "\n";   //--- es el sitio apuntado por pf
  *pf = 100 + *pf;
  cout << "valor de f "  << f << "\n";
  *pf = *pf * 2;
  cout << "valor de f "  << f << "\n";
  return 0;
}
```

Ejemplo 2:

```
1   #include <iostream>
2   using namespace std;
3
4   int main()
5   {
6     float g=4500.99;
7     float *p= nullptr;
8
9     cout << " p " << p << "\n";
10    p= &g;
11    cout << " p " << p << "\n";
12    cout << " p " << &p << "\n";
13
14    *p = *p + 2000;
15    *p = *p * 2;
16    cout << " g "  << g << "\n";
17    return 0;
18  }
```

Ejemplo 3:

```
1   //--- Halla el area y el perimetro de un rectangulo
2   //--- Pero se usara el Heap para almacenar el largo y el ancho
3
4   #include <iostream>
5   using namespace std;
6
7   int main()
8   {
9     float *pAncho= nullptr, *pLargo= nullptr;
10
11    pAncho = new float;
12    pLargo = new float;
13    cout << " Largo : ";
14    cin >> *pLargo;
15    cout <<" Ancho : ";
16    cin >> *pAncho;
17
18    cout << "El Area es : "  <<   *pLargo * *pAncho << "\n";
19    cout << "El perimetro es : " <<  2* *pLargo  +  2* *pAncho;
20    delete pAncho;
21    delete pLargo;
22    pAncho = nullptr;
23    pLargo = nullptr;
24    return 0;
25  }
```