

CS1112: Programación II

Unidad 3: Punteros

Sesión de Laboratorio - 4A

Profesores:

María Hilda Bermejo mbermejo@utec.edu.pe

Jonathan Silva jsilva@utec.edu.pe

Jorge Villavicencio jvillavicencio@utec.edu.pe

Henry Gallegos hgalegos@utec.edu.pe

Ian Paul Brossard ibrossard@utec.edu.pe

Jose Chavez jchaveza@utec.edu.pe

Wilder Nina wnina@utec.edu.pe

José Fiestas jfiestas@utec.edu.pe

Material elaborado por:

Maria Hilda Bermejo, Ruben Rivas



Índice:

- **Unidad 3: Punteros**
 - Definicion
 - Manejo de memoria

3.1

Unidad 3: Punteros, definición y usos



Logro de la sesión:

Al finalizar la sesión, los alumnos podrán:

- Desarrollar programas utilizando punteros para el acceso a valores por referencia.

Ejemplo 1:

Escriba un programa, que permita leer como datos la nota de la PC1 y un puntaje, el cual se adicionará a la nota, a manera de una bonificación. Para realizar el cálculo de la nueva nota, se utilizará una función que reciba como parámetros: la nota y la cantidad de puntos que se le adicionarán o restarán.

Calcule e imprima la nueva nota. Utilice Pase de Referencia con Punteros.

Ejemplo:

Ingrese nota PC1: 13.4

Ingrese puntos: 2.5

Resultado = 15.9

Ejemplo 1: Solución

A continuación se proporciona el programa principal y la función.

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

void corrigeNota(float* notaPtr, float puntos) {
    *notaPtr = *notaPtr + puntos;
}

int main() {
    float notaPc1; float puntos;
    cout<<"Ingreso nota PC1: ";
    cin>>notaPc1;
    cout<<"Ingreso puntos : ";
    cin>>puntos;

    corrigeNota(&notaPc1, puntos);
    cout<<"Resultado = "<<notaPc1<<"\n";
    return 0;
}
```

Ejemplo 2:

Desarrolle un programa que permita realizar un seguimiento de los puntajes de dos equipos en competencia. Los equipos se identifican como Azul y Rojo. El equipo ganador será el que al final del día logre acumular la mayor cantidad de puntos en diversas actividades deportivas. A continuación se proporciona el encabezado de las funciones.

```
void actualizaPuntosAzul(int *ptosAzul)
void actualizaPuntosRojo(int *ptosRojo)
void imprimeTablaPuntos (int *ptosAzul, int *ptosRojo )
```


Ejemplo 2:

SALIDA DEL PROGRAMA:

```
MENU
----
1.INGRESA PUNTOS AZUL
2.INGRESA PUNTOS ROJO
3.IMPRIME PUNTAJES
4.SALIR

Ingrese opcion (1-4):1
Ingrese puntos Equipo Azul:45
```

```
MENU
----
1.INGRESA PUNTOS AZUL
2.INGRESA PUNTOS ROJO
3.IMPRIME PUNTAJES
4.SALIR

Ingrese opcion (1-4):2
Ingrese puntos Equipo Rojo:65
```

```
MENU
----
1.INGRESA PUNTOS AZUL
2.INGRESA PUNTOS ROJO
3.IMPRIME PUNTAJES
4.SALIR

Ingrese opcion (1-4):3
.
TABLA DE PUNTAJES
-----
PRIMER PUESTO: EQUIPO ROJO con 65 puntos...
SEGUNDO PUESTO: EQUIPO AZUL con 45 puntos...
```


Ejercicio 1:

Elabore un programa que utilice una función que calcule el factorial mediante iteraciones. Utilice 4 punteros. Cada puntero apuntará a un entero positivo distinto (4 enteros los ingresa el usuario). Uno de los encabezados de la función sería:

```
int factorial (int &numero)
```

Ejercicio 1 (cont.):

En el programa principal, llame a la función varias veces. En cada llamada enviará:

- El primer entero (pase por valor)
- Un puntero al segundo entero.
- Un pase mediante desreferenciación del tercer entero mediante su puntero.
- Una referencia al tercer puntero que apunta al cuarto entero
- Haga que el cuarto puntero apunte al mismo valor entero que el tercer puntero y envíelo a la función como parámetro.

** Desarrolle nuevos encabezados para la función donde corresponda.*

Ejercicio 2:

Escribe un programa que incluya las siguientes declaraciones:

- a) Un puntero a un carácter
- b) Un puntero a un puntero a carácter
- c) Un puntero a un puntero a entero
- d) Inicializa cada uno de los anteriores e imprime sus valores y también sus direcciones.

Ejercicio 3:

Elabora un programa donde existan dos punteros a 2 números enteros distintos que ingresó el usuario. Manipulando los enteros a través de los punteros, indica para cada uno de ellos si es par o impar.

Ejercicio 4:

Escribe una función calculadora que reciba el operador como char y dos números enteros como operandos. Utilice int* como el tipo de los argumentos.

Ejercicio 5:

Escriba una función que recibiendo los segundos totales de dos horas distintas (inicio y fin), nos informe cuántas horas minutos y segundos se encuentran entre ambas (hh:mm:ss). Utilice punteros para el tratamiento de los números y envíe a la función.

Ejercicio 6:

Elabore un programa que genere un arreglo estático con 30 números aleatorios entre 1 y 50.

Luego debe imprimir los números generados y la cantidad de números mayor que 25.

```
Los valores aleatorios generados son:
```

```
2, 33, 26, 31, 4, 11, 3, 21, 1, 5, 10, 31, 47, 21, 20, 9, 6, 31, 49, 35, 19, 24, 45, 22, 28, 31, 16, 19, 36, 48,
```

```
Los numeros mayor que 25 son: 13
```


Ejercicio 7:

Elabore una función que reciba un puntero “p” como parámetro y actualice el valor de la variable a la que apunta p, a su cuadrado perfecto.

Ejercicio 8:

Elabore una función que reciba un *array* (se debe usar punteros) y le haga una operación de *shift*.

shift: Mover todos los elementos a la derecha o izquierda.

original: [1, 2, 3, 4]

shift izquierda: [2, 3, 4, 1]

Ejercicio 9:

Elabore una función que reciba un *array* (se debe usar punteros) y lo invierta.

original: [1, 2, 3, 4]

invertido: [4, 3, 2, 1]

Resumen

En esta sesión aprendiste:

1. Acceder a la dirección de una variable por medio del **operador referencia &**
2. La sintaxis para definir una variable del tipo puntero.
3. Acceder al valor almacenado en la dirección de memoria almacenada en el puntero por medio del **operador desreferencia ***
4. Utilizar los punteros como parámetros para acceder **por referencia** al valor de una variable fuera del ámbito de una función.

Bibliografía:

Deitel. P.J. and Deitel. H. M. (2016) C++ How to Program, Prentice Hall.

Stroustrup, Bjarne (2013). The C++ Programming Language, 4th Addison-Wesley.

Eckel, Bruce, 2000. Thinking in C++, Vol 1: Introduction to Standard C++, 2nd Edition, Prentice Hall

¡Nos vemos en la siguiente
clase!

