



Curso de C & C++

PADTS en Seguridad 2020

Carlos López. CTS-Cinvestav

Introducción al curso

- Motivación del curso
 - Repaso para quienes ya tienen un cierto nivel de conocimiento de C
 - Preparación para los cursos de programación sobre seguridad



Introducción al curso

- Descripción general del curso
 - Es intensivo
 - Como es intensivo, es mayormente práctico
 - Tiene tres secciones principales: C, C++, Qt en 20 clases de 2 horas
 - Sección C Es solo como antecedente a C++
 - No explica programación o programación estructurada explícitamente
 - Sección C++ Explica fundamentos teórico
 - Explica el paradigma de POO y modelado, no sólo C++.
 - Es mucho más extenso y denso que esta parte de lenguaje C
 - Sección Qt Es una biblioteca para interfaces de usuario
 - Debe ser una sección muy práctica



Introducción al curso

- Reglas y normas generales para el curso
 - Criterios de calificación
 - Tareas Algunas por equipos. Por Teams o Moodle (TBD)
 - Exámenes Son de tarea. Por Teams o Moodle (TBD)
 - Otros criterios generales de PADTS
 - Nota: Los porcentajes no están aún definidos
 - Desconexiones de red
 - Emails. Incluyan PADTS2020 en el subject
 - El chat del curso es solo para información o preguntas generales



Introducción al curso

- Orientación de esta parte del curso
 - Será práctico
 - Para cada sección
 - Prácticas de programación Ej. i para índices, temp para temporal, “do something”
 - Estándares de codificación Ej. siempre break en cada case de un switch
 - Revisión de fallas comunes Ej. == vs =
 - Prácticas que evitar Ej. números mágicos, código complejo y debug
 - Depuración Ej. constantes en vez de macros
 - Es una introducción
 - Se intentará que sea eficiente para que les sea útil
 - Se actualizará presentación por la tarde
 - PREGUNTEN. SI TIENEN DUDAS DE UN CONCEPTO O QUIEREN CONOCER MÁS.



Introducción al curso

- Temario
 - La estructura de un programa en C/C++
 - Variables
 - Display/Entrada funciones especiales
 - Operadores
 - Condicionales
 - Ciclos
 - Funciones
 - Apuntadores
 - Arreglos y cadenas
 - Estructuras y uniones



Introducción al curso

- Bibliografía

- Stroustrup, Bjarne. The C++ Programming Language (Third Edition). Addison-Wesley. ISBN 0-201-88954-4
- Miles, Russ; Hamilton, Kim. Learning UML 2.0. O'Reilly Media, Inc. ISBN-13: 978-0-596-00982-3
- Joyanes Aguilar, Luis. Programación en C++. Algoritmos, estructuras de datos y objetos. McGraw-Hill/Interamericana de España S.A.U. ISBN: 84-481-4645-X
- Brian W. Kernighan / Dennis M. Ritchie. The C Programming Language, Second Edition. Prentice Hall Inc. ISBN 0-13-110362-8.
- Weiss, Mark Allen. Efficient C Programming: a practical approach. Prentice Hall, ISBN 013362658X



1 La estructura de un programa en C/C++



La estructura de un programa en C/C++

- Primer programa

- `#include <stdio.h>`

- `/* Programa que imprime "Hello world".*/`

- `int main(void)`

- `{`

- `printf("Hello world!\n");`

- `return 0;`

- `}`



2 Variables



Variables. Tipos (1 de 3)

- Las variables almacenan información
- En C los tipos de variables son muy importantes. Las variables se tienen que especificar antes de usarlas (no como en Python por ejemplo)
- Existen tipos básicos
 - char, int, float, void
- Modificadores de tipos. Cambian el tamaño e interpretación de las variables
 - short, long, unsigned, signed
- Calificadores de tipos. Cambian las propiedades de las variables. (Prox. slide)
 - const, volatile
- Tipos derivados
 - estructuras, arreglos, apuntadores, uniones



Variables. Tipos (2 de 3)

- Calificadores de tipos de variables
 - `const`
 - `volatile`
- Ámbito de variables
 - `local`
 - `global`
 - `static`
 - `register`
 - `extern`



Variables. Tipos (3 de 3)

- Desde el standard C99 se crearon otros tipos más convenientes para el manejo de enteros, sobre todo para portabilidad en sistemas embebidos. Con estos tipos se asegura el tamaño de las variables.
 - `int8_t` Entero de 8 bits
 - `int16_t` Entero de 16 bits
 - `int32_t` Entero de 32 bits
 - `int64_t` Entero de 64 bits (solo si la plataforma lo soporta)
- También existen los correspondientes tipos sin signo



Variables. Nombres de variables

- No se pueden usar las 32 palabras clave del lenguaje (keywords)
 - while, for, return, switch, case, extern, int, struct, volatile...
- No pueden empezar con dígitos; se permite el guión bajo
- Notación Húngara
 - La inició Charles Simonyi en Xerox y se popularizó en Microsoft
 - c para character, i entero, f flotante, d double, p pointer, u para enteros sin signo
 - Ejemplos:
 - u8_Edad, u8MesaAtendida
 - f_CosAlfa



Variables. Inicialización

- Las inicialización de variables se hace en la declaración
 - `int iVariableEntera = 5000;`
- Para otros tipos de variables no es tan simple
 - Después vemos, en su momento, como se inicializa un arreglo, un arreglo de varias dimensiones (como una matriz), una estructura, un apuntador, etc.
- Tipos de constantes en otras bases numéricas:
 - Hexadecimal: `0x3FC` Binario: `0b00001000`
- En C++ se acostumbra inicializarlas dentro del constructor de un objeto, que es una función que se llama automáticamente al declarar un objeto



Variables. Memoria dinámica

- Requiere el conocimiento de apuntadores
- Se verá después en este mismo curso



Variables. Macros

- Sustituciones de código para mejorar la legibilidad
 - El preprocesador hace la sustitución “as is” (con excepciones)
- Es una “directiva del preprocesador” igual que lo es “#include”
- Normalmente se usan mayúsculas
- Para definir constantes
 - `#define MAX_BUFFER_SIZE (2048)`
 - `#define MAX_BUFFER_SIZE MSG_HEADER_SIZE + MSG_BODY_SIZE`
- y para sustituciones de código (funciones macro)
 - `#define SQUARE(x) ((x)*(x))`



Variables. Coding standards

- SEI y CERT de Carnegie Mellon. Los dioses en ingeniería de software
 - <https://wiki.sei.cmu.edu/confluence/>
 - <https://www.sei.cmu.edu/>
 - <http://www.cert.org>
- MISRA
 - <https://www.perforce.com/resources/qac/misra-c-cpp>
 - **Rule 1.4 Emergent language features shall not be used**
 - *“C99 was finalized in 2011. But it wasn’t until 2013 that MISRA C: 2012 updated the standard to C99. And MISRA C++ — published in 2008 — marked the update to the 2003 version of C++.”*
- AUTOSAR
 - Investigación de tarea



3 Entrada/Salida



Entrada/Salida. printf, scanf y getchar

- Entrada y salida con formato y entrada de caracteres
- printf es una función muy compleja que convierte todo a texto para imprimirlo en pantalla (ok, no necesariamente en pantalla, sino en stdout, standard output)
- Declaración
 - `int printf(char *format, arg1, arg2, ...)`
- Ejemplo
 - `printf("Dirección: 0x%04X %05.10f", i16Address, fKilometros);`
- Secuencias de escape
- Son demasiadas opciones para verlas todas. Investigar de tarea

