Universidad Centroamericana "José Simeón Cañas"

Facultad de ingeniería y arquitectura

Programación de Estructuras Dinámicas

Título:

Guía de estudio para parcial 01

Antiguo Cuscatlán, 9 de septiembre del 2022

Parte Teórica

Estructura de los programas escritos en C++

Explicar línea por línea la estructura del siguiente código, y que hace cada una de las líneas

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

int main() {
  cout << "Hello World!" << endl;
  return 0;
}
```

Funciones

- ¿Cuál es la diferencia entre *declarar* y *definir* una función?
- ¿Qué son parámetros?
- ¿Cómo se usan y qué tipo de parámetros puede usarse en una función?

Arreglos y Cadenas

- ¿Qué son los arreglos?
- ¿Qué tipos de arreglos existen?
- ¿Cúal de las siguientes formas de declarar un arreglo es la correcta?

```cpp
int array01 [] = {1,2,3,4};
int array02 [4] = {1 2 3 4};
int array03 [4] = {1,2,3,4};
int array04 {} = [1,2,3,4];
int array05 [4] = {1,2,3,4};
```

- ¿Qué es un string (Cadena)? ¿Existe alguna diferencia entre los string y los arreglos de caracteres?

Estructuras de datos

- ¿Qué són las estructuras de datos?
- ¿Para qué sirve la palabra reservada typedef y cual es su uso dentro de los códigos?

Punteros simples

- ¿Qué son los punteros?
- ¿Cómo se declaran los punteros?
- ¿Qué es la desreferencia? ¿Y la indirección?

Punteros dobles

- ¿Qué son los punteros dobles?
- ¿Cómo se declara un puntero doble?
- ¿Cómo se hace la referencia con un puntero doble?

Funciones recursivas

- ¿Qué es la recursividad?
- Explique las condiciones necesarias para determinar si una función es recursiva.
- ¿Qué tipos de recursividad existen?

Parte Práctica

1 - John Doe es el jefe de una empresa estadounidense de gestión de paquetes, llamada DeFex. John está buscando realizar un nuevo programa para tener un mejor registro de los paquetes en cada bodega, por lo que le solicita escribir un programa de prueba que tenga una lista de todos los paquetes en el almacén (10).
Cada paquete tiene:

- ID (Número del 1 al 100 generado aleatoriamente)
- Es frágil (Sí/ No)
- Peso en Kg
- Tamaño del paquete (Pequeño / Mediano / Grande)

El programa debe permitir, listar todos los paquetes (ID), ver información detallada de cada paquete (Ingresar ID y mostrar toda la información), y modificar cualquiera de los registros.

2 - Ingresar un número y mostrar su equivalente en binario usando una función recursiva.

3- Realice un programa que cumpla con las siguientes características

1. Cree un programa que contenga una estructura de tipo [estudiante] que deberá almacenar: nombre y edad
2. Inicialice los valores de la estructura a los que sean de su preferencia
3. Cree un puntero que apunte a la dirección de memoria de la estructura recién creada
4. Cree un puntero doble que apunte a el puntero simple creado en el paso 3
5. Utilizando el puntero simple modifique los valores previamente almacenados en la estructura
6. Muestre los valores almacenados, utilizando una función que reciba la estructura Estudiante
7. Utilizando una función que utilice el puntero doble cambie la edad del registro
8. Utilizando una función que utilice el puntero simple cambie el nombre del registro
9. Muestre los valores almacenados, utilizando una función que reciba el puntero simple.
10. Intente modificar la edad del registro usando una función que reciba la estructura Estudiante (no punteros)
11. Compruebe que ningún cambio fue hecho en el registro

Opcionalmente: Utilizar typedef

Solución:

Estructura, uso de typedef y declaración de funciones a utilizar.

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

struct estudiante {
    string nombre;
    int edad;
};
typedef struct estudiante Estudiante;

void mostrarEstudiante(Estudiante unEstudiante);
void mostrarEstudiantePuntero(Estudiante *puntero);
void cambiarEdad(Estudiante **punteroDoble, int edad);
void cambiarNombre(Estudiante *puntero, string nombre);
void cambiarEdadSinPuntero(Estudiante estudiante, int edad);
```

Estructura principal del programa

```cpp
int main() {
    // 1 y 2. Creando un registro de estudiante y inicializando sus datos
    Estudiante carlos;
    carlos.nombre = "Carlos";
    carlos.edad = 20;

    // 3 y 4
    Estudiante *puntero = &carlos;
 // Creamos un puntero que apunte a la direccion de memoria de la struct
    Estudiante **punteroDoble = &puntero;
 // Puntero doble que apunta al puntero simple
    // punteroDoble -> puntero -> Estudiante (carlos)

    // 5. Modificando valores de la estructura usando el puntero simple
    cout << "Modificando nombre...\n";
    (*puntero).nombre = "Carlos Mercado";
    cout << "Modificando edad..\n";
    (*puntero).edad = 21;

    // 6. Mostrando valores actuales de la estructura
    cout << "Accediendo al registro:\n";
    mostrarEstudiante(carlos);

    // 7 y 8. Modificando valores de la estructura usando el puntero doble
    cout << "Modificando edad y nombre...\n";
    cambiarEdad(punteroDoble, 22);
 // Modificamos la edad con una funcion que recibe un puntero doble y la edad
    cambiarNombre(puntero, "Carlos");
 // Modificamos el nombre con una funcion que recibe un puntero simple y el nombre

    // 9. Mostrando valores actuales de la estructura
    cout << "Accediendo al registro:\n";
    mostrarEstudiantePuntero(puntero);

    // 10 y 11
    cambiarEdadSinPuntero(carlos, 19);
    cout << "Accediendo al registro:\n";
    /* La edad no sera 19 por que no modificamos la estructura carlos
    sino una copia de ella que fue enviada a la funcion cambiarEdadSinPuntero()
    */
    mostrarEstudiantePuntero(puntero);

    return 0;
}
```

Definición o implementación de las funciones previamente declaradas.

```
void mostrarEstudiante(Estudiante unEstudiante) {
    cout << "Nombre: " << unEstudiante.nombre << ", Edad: " <<
unEstudiante.edad << "\n\n";
}

void mostrarEstudiantePuntero(Estudiante *puntero) {
    cout << "Nombre: " << (*puntero).nombre <<  ", Edad: "
 << (*puntero).edad << "\n\n";
}

void cambiarEdad(Estudiante **punteroDoble, int edad) {
    (**punteroDoble).edad = edad;
}

void cambiarNombre(Estudiante *puntero, string nombre) {
    (*puntero).nombre = nombre;
}

void cambiarEdadSinPuntero(Estudiante estudiante, int edad) {
    estudiante.edad = edad;
}
```