ARELLANO GRANADOS ANGEL MARIANO
218123444

Punteros

Objetivo.- Aprender los conceptos básicos de punteros, así como su clasificación y forma de estructuras.

Investigación:

Defina que es un puntero?

Las variables apuntadores contienen direcciones de memoria como sus valores. Por lo general, una variable contiene directamente un valor específico. Sin embargo, un apuntador contiene la dirección de memoria de una variable que, a su vez, contiene un valor específico.

Defina que es puntero paso por referencia?

En C++, los programadores pueden usar apuntadores y el operador indirección (*) para realizar el paso por referencia. Cuando se llama a una función con un argumento que se debe modificar, se pasa la dirección del argumento. Por lo general, para realizar esto se aplica el operador dirección (&) al nombre de la variable cuyo valor se va a modificar.

EJEMPLO:

```
1  // Fig. 8.7: fig08_07.cpp
2  // Uso del paso por referencia con un argumento apuntador para elevar
3  // al cubo el valor de una variable.
4  #include <iostream>
5  using std::cout;
6  using std::endl;
7
8  void cuboPorReferencia( int * ); // prototipo
9
10 int main()
11 {
12    int numero = 5;
13
14    cout << "El valor original de numero es " << numero;
15
16    cuboPorReferencia( &numero ); // pasa la dirección de numero a cuboPorReferencia
17
18    cout << "\nEl nuevo valor de numero es " << numero << endl;
19    return 0; // indica que terminó correctamente
20 } // fin de main
21
22 // calcula el cubo de *nPtr; modifica la variable numero en main
23 void cuboPorReferencia( int *nPtr )
24 {
25    *nPtr = *nPtr * *nPtr * *nPtr; // eleva *nPtr al cubo
26 } // fin de la función cuboPorReferencia
```

```
El valor original de numero es 5
El nuevo valor de numero es 125
```

ARELLANO GRANADOS ANGEL MARIANO
218123444

Defina que es un puntero paso por valor?

Cuando pasamos un puntero como parámetro de una función por valor pasa lo mismo que con cualquier otro objeto.

Dentro de la función trabajamos con una copia del parámetro, que en este caso es un puntero. Las modificaciones en el valor del parámetro serán locales a la función y no se mantendrán después de retornar.

Sin embargo, no sucede lo mismo con el objeto apuntado por el puntero, puesto que en ambos casos será el mismo, ya que tanto el puntero como el parámetro tienen como valor la misma dirección de memoria. Por lo tanto, los cambios que hagamos en los objetos apuntados por el puntero se conservarán al abandonar la función.

EJEMPLO:

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

void funcion(int *q);

int main() {
   int a;
   int *p;

   a = 100;
   p = &a;
   // Llamamos a funcion con un puntero
   funcion(p); // (1)
   cout << "Variable a: " << a << endl;
   cout << "Variable *p: " << *p << endl;
   // Llamada a funcion con la dirección de "a" (constante)
   funcion(&a);  // (2)
   cout << "Variable a: " << a << endl;
   cout << "Variable *p: " << *p << endl;

   return 0;
}

void funcion(int *q) {
   // Cambiamos el valor de la variable apuntada por
   // el puntero
   *q += 50;
   q++;
}
```

ARELLANO GRANADOS ANGEL MARIANO
218123444

Cuál es la diferencia entre puntero paso por valor y paso por referencia?

Al ver ambas definiciones concluyo que la diferencia esta en que cuando usamos paso por referencia se puede modificar el valor del parámetro y el puntero en sí, es decir la dirección de memoria; En cambio con el paso por valor no se puede modificar el parámetro pero si cambiare la dirección de memoria tras retornar al main.

Referencias:

- http://conclase.net/c/curso/cap15#:~:text=Cuando%20pasamos%20un%20puntero%20como,que%20con%20cualquier%20otro%20objeto.&text=Sin%20embargo%2C%20no%20sucede%20lo,la%20misma%20direcci%C3%B3n%20de%20memoria.
- DEITEL, HARVEY M. Y PAUL J. DEITEL. Cómo programar en C++. Sexta edición, PEARS ON EDUCACIÓN, México 2008.