Laboratorio de Computación II

Profesor: Angel Simón
JTP: Brian Lara

Contenidos

Parcial 1

- Estructuras
- Archivos
- Asignación dinámica de memoria

Parcial 2

- Programación orientada a objetos
  - Atributos y métodos
  - Constructores y destructores
  - Herencia
  - Sobrecarga
  - Polimorfismo

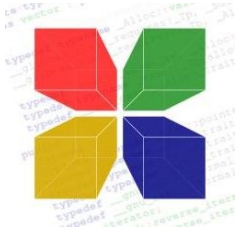
Final = Parcial 1 + Parcial 2

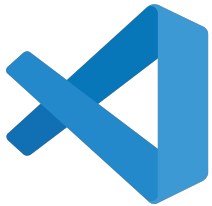
Evaluaciones

- Parcial 1
  - Recuperatorio: Primer fecha de final
- Parcial 2
  - Recuperatorio: Segunda fecha de final
- Final

Todas las evaluaciones son proyectos de software.

Software





Clasificación de variables

- Por tipo de dato
- Por dimensión
- Por alcance

Tipos de datos

Números enteros	Números reales	Texto

Por dimensión

Variables simples

```
int edad;

char caracter;

float precio;

bool confirmar;
```

Vectores

```
int vCant[50];

char frase[100];

float pagos2019[12];

bool vec[5];
```

Matrices

```
int mNotas[5][50];

char nombres[10][50];

float ingresos[12][5];

bool confirmar[3][3];
```

Por alcance - Global

```
int numero;

int main(){
  numero=5;
}

int miFuncion(){
  numero = 10;
}
```

En la mayoría de los casos. No es una buena práctica.

Por alcance - Local / Función

```
void miFuncion(float var3){
    int var;
    char var2;
    /* var, var2, var3 son variables locales a miFuncion.
No existen fuera */
}

int main(){
    int var;
    /* var es local a main y no tiene relación con la de
miFuncion */
}
```

Por alcance - Ámbito de llaves

```
int main(){
    int var;
    var = 0;
    if (var == 0){
        int var2;
        var = 1;
        var2 = 1; //OK
    }
    var2 = 0; // ERROR
}
```

Estructuras de programación

- Estructura secuencial
- Estructura de decisión
- Estructura de repetición

Estructura de decisión

```
if (condición){
  /* Instrucciones
por verdadero*/
}
else{
  /* Instrucciones
por falso */
}
```

Decisión simple

```
switch (var){
  case 1:
    /* Instrucciones cuando var == 1 */
  break;
  case 10:
    /* Instrucciones cuando var == 10 */
  break;
  default:
    /* Instrucciones cuando no se cumple
ningún caso (es opcional)*/
  break;
}
```

Decisión mútiple

Estructura de repetición - Ciclo exacto

```
for (i = 1; i <= 10; i++){
  /* Repite las instrucciones 10 veces.
En cada iteración i aumenta de a uno */
}
```

```
for (i = 1; i <= 10; i=i+2){
  /* Repite las instrucciones 5
veces. En cada iteración i aumenta de
a dos */
}
```

```
for (i = 10; i >= 1; i--){
  /* Repite las instrucciones 10 veces.
En cada iteración i disminuye de a uno
*/
}
```

Estructura de repetición - Ciclo inexacto

```
while (condicion){
  /* Repite las instrucciones mientras condicion sea
verdadera. Puede nunca ser verdadera. Se debe garantizar que
la condición pueda ser falsa alguna vez. */
}
```

```
do{
  /* Repite las instrucciones mientras condicion sea verdadera.
El ciclo se ejecuta al menos una vez. Se debe garantizar que la
condición pueda ser falsa alguna vez. */
}
while (condicion);
```

Punteros

Punteros (ejemplo)

```
int main(){
   int a;
   a = 5;

   int *b;
   b = &a;
   *b = 10;
   cout << *b;
   return 0;
}
```

Punteros (ejemplo explicado)

```
int main(){
   int a;
   a = 5;

   int *b;
   b = &a;
   *b = 10;
   cout << *b;
   return 0;
}
```

Se declara un puntero a entero llamado b

Se le asigna a b la dirección de memoria de a

En la parte de la memoria donde apunta b se asigna el valor 10 (o sea, en a)

Se muestra el contenido de donde está apuntando b. O sea 10.

Funciones

```
void funcion(int *a,
int *b){
  int c;
  c = *a;
  *a = *b;
  *b = c;
  return;
}
```

```
int funcion2(int a,
int b){
  int max = a;
  if (b > max){
    max = b;
  }
  return max;
}
```

Break de 30 minutos

Ejercicios para hacer en el break

1) Dado un vector de enteros de 10 elementos. Hacer una función llamada obtenerMaximo que reciba el vector y el tamaño y devuelva el mayor valor del vector.

2) Un empresa dispone de 15 productos que se comercializan en 10 sucursales. Hacer un programa que permita cargar las ventas realizadas el mes pasado. Por cada venta se registra: Día de la venta, Código de producto (1 a 15), Código de sucursal (1 a 10) e importe de la venta.
El fin de la carga de datos se indica con un día de la venta igual a cero.
Calcular e informar:

- Por cada producto y sucursal, el total facturado.
- Las sucursales sin ventas el mes pasado.
- El producto que haya registrado la mayor cantidad de ventas.