Universidad de Los Andes
Escuela de Ingeniería de Sistemas
Departamento de Computación

PROGRAMACIÓN 2

Clase 03

Junior Altamiranda
altamira@ula.ve

AGENDA

- Asignación dinámica de memoria con new y delete.
- Ejercicios

Asignación dinámica con los operadores new y delete

Hay dos formas princiales por medio de las cuales un programa puede almacenar información en la memoria del computador.

- La primera es por medio del uso de variables, en cuyo caso la cantidad de memoria necesaria queda fijada en tiempo de compilación y no puede ser cambiada durante la ejecución del programa. Esto es, si se ha reservado memoria para un vector de 10 elementos de tipo int, este vector no puede cambiar durante la ejecución

- La segunda forma permite signar memoria en tiempo de ejecución según se vaya necesitando.

Asignación dinámica con los operadores new y delete

C++ tiene 2 operadores para la asignación dinámica de memoria: new y delete. Sus formas generales son:

var_puntero = new tipo_variable;

delete var_puntero;

Asignación dinámica con los operadores new y delete

El operador new reserva memoria dinámica de cualquier tipo de datos:

tipos básicos (int, float, char, etc)
tipos definidos por el usuario (clases o estructuras).

```
float *var1 = new float;
*var1 = 4.5;
delete var1;
```

Asignación dinámica con los operadores new y delete

Usasndo new para un vector, el tamaño del vector se sitúa entre corchetes. Con delete el contenido es borrado.

```
var_puntero = new tipo_variable [n];
Delete [] var_puntero;
```

Si existe insuficiente memoria el operador new devuelve NULL

```
int n=100;
int *datos = new int[n];
for (int i = 0; i<n;i++){
  datos[i]=i+2;
};
delete []datos;
```

Asignación dinámica con los operadores new y delete

Asignación dinámica: Reserva de espacios de memoria en tiempo de ejecución.
Sintaxis:

```
<tipo_x> *ptr_;
ptr_ = new <tipo_x>;
delete ptr_;
```

new:
- Crea un objeto de tamaño adecuado
- Llama al constructor del objeto
- Devuelve un apuntador del tipo indicado
- Si no hay memoria disponible, retorna 0

delete:
- Libera el espacio ocupado por el objeto
- Llama al destructor del objeto

Asignación dinámica con los operadores new y delete

Para tipos básicos de datos:

```
float *ptrFloatX = new float(2.71);
delete ptrFloatX;
```

Para un arreglo:

```
int *ptrAr = new int[40];
delete [] ptrAr;
```

(No olvidar el [])

Ejemplo de asignación dinámica de memoria

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

int main() {
    int n;

    cout << “¿Cuántos elementos?”;
    cin >> n;

    int *ptrNotas = new int[n];
    delete [] ptrNotas;
}
```

Ejercicio 1

Realizar un programa en C++ que almacene las calificaciones de los alumnos de una sección y pase este vector a una función que calcule el promedio de las calificaciones. El programa solicitará la cantidad de alumnos que tiene la sección y no desperdiciará espacio de memoria en el arreglo. Es decir, se debe realizar asignación dinámica de memoria al crear el vector.

Ejercicio 2

PDVSA está interesada en llevar las estadísticas de fallas en sus plantas. Realizar un programa en lenguaje C++ que permita, a través de un vector de estructuras, almacenar los campos Ubicación, tipo y fecha, de cada una de las fallas.
La fecha debe ser implementada como una estructura.
Utilice memoria dinámica.
Escribir además una función que permita conocer cuántas fallas tipo 2 se produjeron en enero de 2012.