

Tarea 10

Las preguntas precedidas por un asterísco son para los alumnos de maestría. En licenciatura, dan puntos extras.

Ejercicio 1 *Vimos en la previa clase que el destructor de un objeto es sistemáticamente llamado al salir de un bloque de llaves. Escribir dos ejemplos de programas a través de los cuales se ve:*

1. *que sí el destructor es llamado al salir de un bloque donde está definido el objeto,*
2. *que aún está llamado si salimos de un bloque (por ejemplo, de un ciclo for) por un **goto**.*

Ejercicio 2 *¿Qué declaración tengo que quitar para poder compilar? y ¿por qué?*

```
int tata(double &x);
float tata(const string &s,const double &x);
float tata(double &x);
int  tata(const string &s,const double &x);
int  tata(const string &s,unsigned int k);
int  tata(const double &x);
```

Ejercicio 3 *Re-escribir la clase Queue del ejercicio 2 de la tarea 9 añadiendo constructores sobrecargado:*

1. *un constructor por default que supondrá que los datos son de 4 bytes de largo,*
2. *un constructor con el número de bytes por dato,*
3. *un constructor con el número de bytes por dato, y un primer elemento que añadir a la cola (es decir un apuntador void).*

***Ejercicio 4** *Buscar en la documentación de su elección todos los constructores de string posibles y escribir un programa en que se usará cada uno de ellos.*

Ejercicio 5 *En C++ es posible sobrecargar los operadores usuales, y en particular los operadores sobre flujo, por ejemplo para imprimir el contenido de una estructura en la salida estándar. Típicamente, se escribirá el operador sobre-cargado como una **función** parte (y no un método):*

```
friend ostream& operator<<(ostream& os, const S& s) {
...
}
```

donde S es la clase que se quiere imprimir. Escribir para la clase *Complex* (ver tarea 8 y 9) una función que sobrecarga el operador " \leq " (es decir que imprime el contenido) y dar un ejemplo de su uso.

***Ejercicio 6** *Mostrar a través de un ejemplo que, aunque un método de una clase es const, se puede, jugando con el apuntador `this`, cambiar elementos del objeto dentro de este método.*