Programación 2009

Tarea 11

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

Ejercicio 1 *Tomar la clase Complex de las tareas precedentes, y hacer inline a los métodos para los cuales piensas que vale la pena hacerlo.*

Ejercicio 2 *Escribir un programa de manipulación de imagenes (con una estructura Image), en el cual se accede a los valores de pixeles con métodos de aceso. Verificar empiricamente que, al recorrer imagenes de gran tamaño, es mas eficiente en términos de tiempo escribir esos métodos inline que "normales". Comparar también el tiempo de recorrido con un acceso directo a los elementos de la clase Image.*

Ejercicio 3** *Explicar por qué puede ser complicado para el compilador manejar funciones definidas como inline en un .h, cuando esas funciones contienen una* ***variable local estática*. Verificar si su compilador lo puede manejar bien o no con un ejemplo bien elegido.*

Ejercicio 4 *Sean dos archivos file1.cpp y file2.cpp como sigue :*

```
// Archivo file1.cpp
extern double b;
double a = b * 1;
```

y por otra parte :

```
// Archivo file2.cpp
extern double a;
double b = a + 2;
```

Si compilamos los dos archivos en un mismo ejecutable, ¿ cuál es el problema en este ejemplo, si hay uno?

Ejercicio 5 *Usando las propiedades de control de acceso y la palabra llave static, escribir una clase de que sólo pueda haber una única instancia en todo el programa.*

Ejercicio 6 *Para cada declaración $f1$, $f2$, $f3$, explicar claramente qué tipo de funciones es (valor de regreso y argumentos). Luego, escribir un programa en que se usarán esos métodos.*

```
int (Complex::*f1)(bool);
const Complex &(*f2(const Complex &))(Complex const *);
int (Complex::*f3)(Complex& (Complex::*)(const double&));
```

***Ejercicio 7** *En el programa siguiente, decir cuantas instancias distintas de la clase Image han sido creadas en total (incluyendo las que luego son destruidas) y justificar su cuenta. Luego, verificar esta cuenta implementando una clase Image básica y contando las instanciaciones por una variable static en los constructores (como lo vimos en la clase). ¿Habrán unas optimizaciones hechas por el compilador?*

```
int f1(cont Image &img) {return 0;};
Image f2(int a) {return Image(100,100);}
Image f3(Image img) {Image tmp; return temp;};
Image f4(Image *img) {
    Image tmp(100,100);
    return tmp;
}
int f5(Image img,unsigned int a) {
if (a<=1) return 0;
return f5(img,a-1);
}
int main() {
  Image a = f2(1);
  f2(f1(a)+f5(a,10));
  f1(f3(f4(&a)));
  return 0;
}
```