Unidad Nº2

Preguntas orientadoras

1) Mencione y defina los tipos de ámbitos de las variables que se presentaron en la teoría. ¿Por qué definiría como global una variable de tipo static?.

2) ¿Qué modificador elegiría para indicar que una variable puede cambiar su valor entre accesos aún cuando no pareciera haber sido modificado?¿En qué aplicaciones se utiliza este modificador?

3) ¿Por qué puede ser que el operador new retorna un puntero y no una referencia? (Ayuda: piense en la diferencia entre punteros y referencias para encontrar la respuesta).

4) Indique cuál es la función de los constructores y para qué existen los destructores.

5) Llene los espacios en blanco:

- Se tiene acceso a los miembros de clase vía el operador _____ en conjunción con un objeto de clase o vía el operador _____ en conjunción con un apuntador a un objeto de clase.
- Los miembros de una clase especificados como _____ son sólo accesibles a las funciones miembro de la clase y amigos de la clase.
- Un _____ es una función miembro especial utilizada para inicializar los miembros de datos de una clase.
- El acceso por omisión para los miembros de una clase es _____.
- Los miembros de una clase especificados como _____ son accesibles en cualquier parte en que un objeto de la clase esté en alcance.
- El operador _____ asigna dinámicamente memoria para un objeto de un tipo específico y regresa un _____ a dicho tipo.
- Las operaciones de entrada son soportadas por la clase _____.
- Las operaciones de salida son soportadas por la clase _____.

6) Encuentre el o los errores y corrija:

- void ~Time(int);
- Suponga la siguiente definición parcial de la clase Time

```
class Time
{
public:
//function prototypes
private:
int hour = 0;
int minute = 0;
int second = 0;
};
```

7) ¿Se pueden utilizar los nombres definidos en un espacio de nombres sin utilizar la palabra reservada using?

Ejercicios

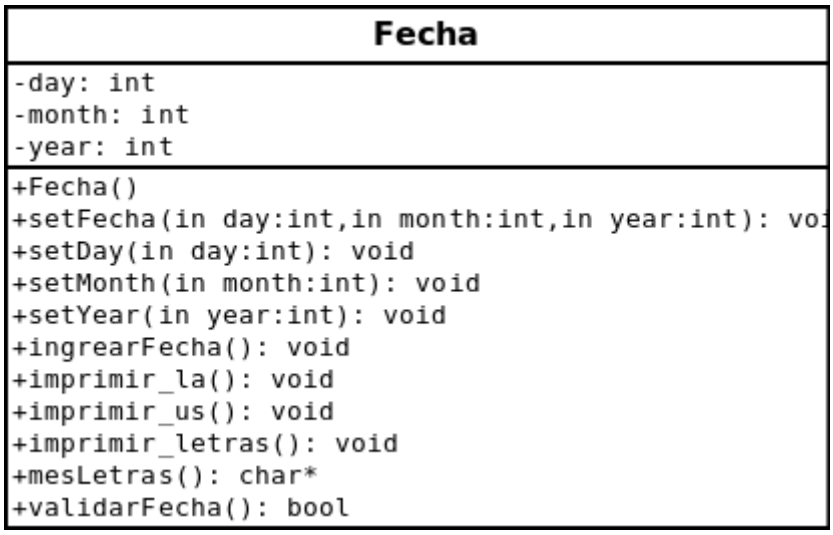
1) ¿Qué inconveniente presenta el programa que se muestra a continuación? Discuta las posibles soluciones.

```
int main (int argc, char *argv[])
{
   string cadena;
   cin >> cadena;
   getline(cin, cadena);
   cout << "La cadena ingresada es: " << cadena << endl;
   return 0;
}
```

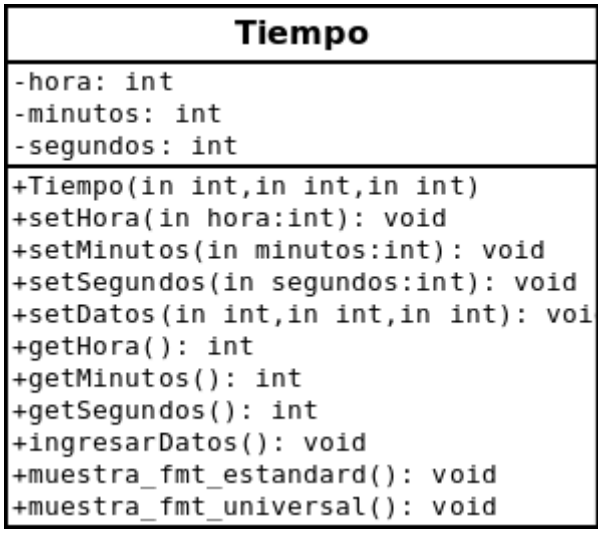
2) ¿Qué está mal en este programa? Mencione 3 formas de corregirlo.

```
#include <iostream>
int main()
{
   cout << "¡Hola, mundo!" << endl;
   return 0;
}
```

3) Crear la clase Fecha con todos los datos miembro y atributos que se muestran en el diagrama de clase siguiente y luego, crear un programa cliente que demuestre el uso de cada una de las funciones.



4) Crear la clase Tiempo con todos los datos miembro y atributos que se muestran en el diagrama que sigue y luego, crear un programa cliente que muestre el uso de cada una de las funciones.



5) El siguiente diagrama muestra muestra un concepto fundamental de la programación orientada a objetos: **La composición**. Implementar la clase Empleado y escribir un programa que imprima en pantalla la siguiente información:

Perez, Juan

Contratado el: 1 de Julio de 1999

Fecha de nacimiento: 31 de Diciembre de 1978

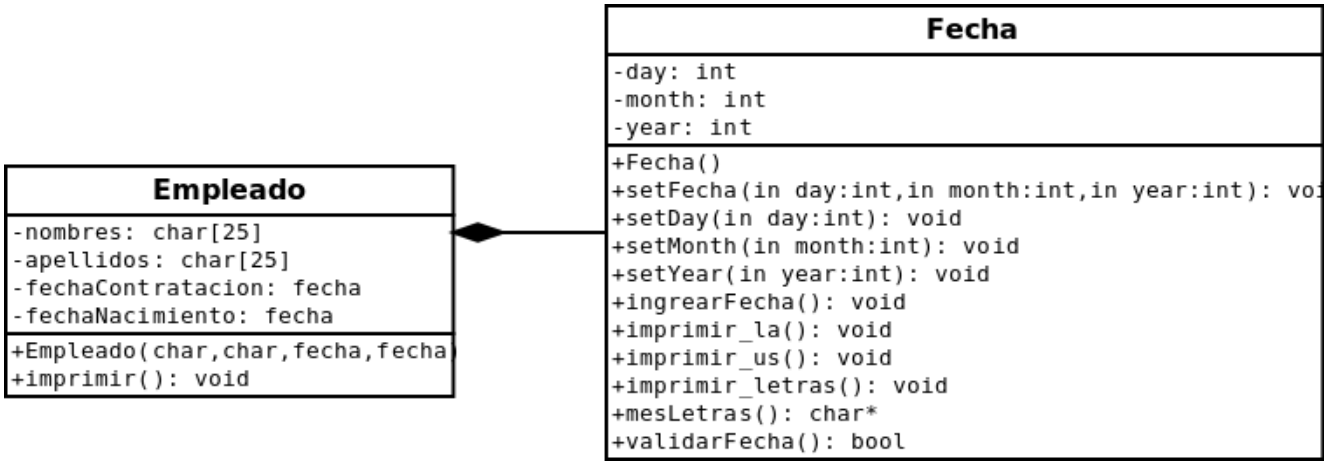
Lopez, Pedro

Contratado el: 1 de Julio de 1999

Fecha

de nacimiento: 25 de Febrero de 1980

Presione una tecla para continuar . . .



6) En una competencia de ciclismo intervienen un número desconocido de deportistas, cada uno realiza dos pruebas, una por tiempo y otra por número de vueltas. Se deben registrar los siguientes datos de cada participante: nombre, fecha de nacimiento, tiempo en la primera prueba y número de vueltas en la segunda prueba, respectivamente. Imprimir en pantalla el nombre del participante que realizó la primera prueba en el menor tiempo, el nombre del participante que hizo la mayor cantidad de vueltas y 5 columnas en las que se muestre para cada participante:

Nombre, edad, tiempo en la primera prueba, número de vueltas en la segunda prueba, diferencia de tiempo respecto del más rápido en la primera prueba y diferencia de vueltas respecto del que hizo más vueltas en la segunda prueba.

Se pide implementar un programa en lenguaje C++, orientado a objetos, que resuelva el problema. Además, se pide el diagrama de clases y de secuencia.

NOTA: En todos los ejercicios, se deben instanciar los objetos en la memoria HEAP.