

1.- Las máquinas de venta como las de bebidas, café, etc., son objetos compuestos por diferentes objetos. Considere una máquina que vende 5 tipos de bebidas en lata (o tarro ~300 cc). Haga un listado de al menos 5 tipos de objetos importantes para el funcionamiento de esta máquina de autoservicio (Ej. depósito de bebidas tipo X).

Para cada tipo de objeto liste al menos 2 operaciones o responsabilidades que puedan ser útiles para crear un programa que simule una de estas máquinas. 20 pts.

2.- Preguntas de respuestas corta: (4 pts c/u).

- ¿Qué es una función sobrecargada?
- ¿Cómo el compilador distingue entre dos funciones sobrecargadas?
- ¿Qué funciones pueden acceder a miembros dato declarados como protegidos en una clase?
- ¿Cuál es la diferencia entre las siguientes declaraciones:

```
void foo(const int * p) { .. };
void foo(int * const p) { .. };
```
- Mencione dos ventajas de los lenguajes orientados a objeto por sobre los no orientados al objeto.
- ¿Qué elementos del lenguaje C++ son utilizados cuando necesitamos que la ligazón del nombre de una función y su código sea efectuada en tiempo de ejecución?
- En proyectos de software pequeños (1 ó 2 personas) ¿en qué radica la dificultad del proyecto?
- En proyectos de software grandes (> 5 personas) ¿en qué radica la dificultad del proyecto?
- ¿Qué se hace en la etapa de **análisis** del proceso de desarrollo de software?
- ¿A qué corresponden los atributos de un objeto?
- ¿Cuál es la diferencia entre miembros protected y miembros private?
- Dé un ejemplo de una plantilla para una función (= ejemplo de una función template).
- ¿Cuándo ocurre una fuga de memoria (memory leak)?

3.- Considere la especificación del registro de desplazamiento 74194 dada:

Function Table

Clear	Mode		Clock	Inputs				Outputs					
	S1	S0		Serial		Parallel		QA	QB	QC	QD		
				Left	Right	A	B					C	D
L	X	X	X	X	X	X	X	X	X	L	L	L	L
H	X	X	L	X	X	X	X	X	X	QA0	QB0	QC0	QD0
H	H	H	↑	X	X	a	b	c	d	a	b	c	d
H	L	H	↑	X	H	X	X	X	X	H	QA _n	QB _n	QC _n
H	L	H	↑	X	L	X	X	X	X	L	QA _n	QB _n	QC _n
H	H	L	↑	H	X	X	X	X	X	QB _n	QC _n	QD _n	H
H	H	L	↑	L	X	X	X	X	X	QB _n	QC _n	QD _n	L
H	L	L	X	X	X	X	X	X	X	QA0	QB0	QC0	QD0

H = High Level (steady state), L = Low Level (steady state), X = Don't Care (any input, including transitions)
 ↑ = Transition from low to high level; a, b, c, d = The level of steady state input at inputs A, B, C, or D, respectively
 QA0, QB0, QC0, QD0 = The level of QA, QB, QC, or QD, respectively, before the indicated steady state input conditions were established.
 QA_n, QB_n, QC_n, QD_n = The level of QA, QB, QC, respectively, before the most recent ↑ transition of the clock.

Cree una clase llamada CregDesplazamiento. Ésta debe encapsular el estado y comportamiento de este registro. 20 pts.

4.- Cree la clase Rectángulo. Ésta debe incluir como atributos dos puntos que definen el rectángulo (x1,y1,x2,y2), el espesor de la línea (1,2,3,4,5, .. puntos) y el color del borde.

Luego cree la clase RectanguloLleno como subclase de la clase anterior. En ésta incluya un campo para el color interno del rectángulo. Implemente el seTraslapan el cual permite determinar si dos rectángulos poseen intersección no nula. Usted defina los parámetros que le parezcan más adecuados. Obs. No se preocupe de otros métodos que completen la caracterización de rectángulos. 20 pts.