

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.

Listado de Objetos importantes:

Objeto	Operaciones
1.- Botón de Selección del producto,	Presionar, ponerEtiqueta
2.- Sistema de ingreso y validación de dinero (monedas/billetes)	AceptarDinero, ValidarDinero
3.- Sistema de cuenta de dinero y calculo de vuelto.	IniciarNuevoIngreso, acumularUltimoIngreso, desplegarIngresoAcumulado, calcularVuelto
4.- Depósito de dinero recaudado	ReseteoDepósito, acumularVenta, EntregaValorRecaudado, HabrirCandado, CerrarCandado, liberarValorVuelto.
5.- Sistema de Alimentación y Luces.	EncenderLuces, EnergizarSistema
6.- Depósito de bebidas del un determinado tipo.	EstaVacío, DesplegarEstado, obtenerNúmeroBebidasVendidas.

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

a) ¿Qué es una función sobrecargada?

Aquella que dentro del mismo alcance posee el mismo nombre que otra función.

b) ¿Cómo el compilador distingue entre dos funciones sobrecargadas?

Por los parámetros, los cuales deben diferir en número y/o tipo.

c) ¿Qué funciones pueden acceder a miembros dato declarados como protegidos en una clase?

Métodos de la misma clase, clase derivadas, y funciones amigas.

d) ¿Cuál es la diferencia entre las siguientes declaraciones:

```
void foo(const int * p) { ..};
```

```
void foo(int * const p) { ..};
```

En el primer caso en foo no puedo cambiar el contenido apuntado por p, pero sí puedo cambiar el valor al cual p apunta. En el segundo caso puedo cambiar el contenido apuntado, pero no el lugar que apuntamos.

e) Menciones dos ventajas de los lenguajes orientados a objeto por sobre los no orientados al objeto.

Permiten una mejor encapsulación de los datos y su comportamiento.

Permiten definir jerarquías de en que los objetos pueden heredar de varias clases

Permiten el ligado dinámico en tiempo de ejecución.

f) ¿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?

Definimos métodos virtuales y creamos un conjunto de clases derivadas.

g) En proyectos de software pequeños ( 1 ó 2 personas) ¿en qué radica la dificultad del proyecto?

En el diseño y desarrollo de algoritmos.

h) En proyectos de software grandes (> 5 personas) ¿en qué radica la dificultad del proyecto?

Administración de detalles y la comunicación entre los equipos de trabajo.

i) ¿Qué se hace en la etapa de **análisis** del proceso de desarrollo de software?

Se estudia y modela el mundo real donde se plantea el problema.

j) ¿A qué corresponden los atributos de un objeto?

Los atributos corresponden a los miembros dato de un objeto. En ellos se almacena el estado del objeto.

k) ¿Cuál es la diferencia entre miembros protected y miembros private?

La diferencia está en que los miembros privados no son visibles para la clases derivadas, las protected si.

l) Dé un ejemplo de una plantilla para una función (= ejemplo de una función template).

```
template <class T>
void swap(T &A, T &B) {
    T t = A;
    A = B;
    B = t;
}
```

m) ¿Cuándo ocurre una fuga de memoria (memory leak)?

Se produce cuando el programa no llama al destructor para todos objetos que han sido creados en el heap con el operador new.

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 <sub>n</sub>	QB <sub>n</sub>	QC <sub>n</sub>
H	L	H	↑	X	L	X	X	X	X	L	QA <sub>n</sub>	QB <sub>n</sub>	QC <sub>n</sub>
H	H	L	↑	H	X	X	X	X	X	QB <sub>n</sub>	QC <sub>n</sub>	QD <sub>n</sub>	H
H	H	L	↑	L	X	X	X	X	X	QB <sub>n</sub>	QC <sub>n</sub>	QD <sub>n</sub>	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<sub>n</sub>, QB<sub>n</sub>, QC<sub>n</sub>, QD<sub>n</sub> = 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.

```
class CregDesplazamiento {
public:
    CregDesplazamiento();
    bool shiftRight(bool rin);
    bool shiftLeft(bool fin);
    void clear();
    void load(bool a, bool b, bool c, bool d);
    bool obtengaQA() const;
    bool obtengaQB() const;
    bool obtengaQC() const;
    bool obtengaQD() const;
private:
    bool qa, qb, qc, qd;
};

bool CregDesplazamiento :: shiftRight( bool rin) {
    bool t = qd;
    qd=qc;
    qc=qb;
    qb=qa;
```

```

        qa=rin;
        return t;
};
bool CregDesplazamiento :: shiftLeft( bool fin) {
    bool t = qa;
    qa=qb;
    qb=qc;
    qc=qd;
    qd=fin;
    return t;
};

void CregDesplazamiento :: clear () {
    qa=qb=qc=qd=false;
}
void CregDesplazamiento :: load (bool a, bool b, bool c, bool d) {
    qa=a;
    qb=b;
    qc=c;
    qd=d;
}

bool CregDesplazamiento :: obtengaQA () {
    return qa;
}

bool CregDesplazamiento :: obtengaQB () {
    return qb;
}
bool CregDesplazamiento :: obtengaQC () {
    return qc;
}
bool CregDesplazamiento :: obtengaQD () {
    return qd;
}

```

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.

```

class Rectangulo {
public:
    Rectangulo ();
    Rectangulo(int xx1, int yy1, int xx2, int yy2);
    bool seTraslapan (const Rectangulo &r) const;
// Otros métodos....
private:

```

ELO 326

Primer Certamen Martes 2 de octubre de 2001

```
    int x1,y1,x2,y2;
    int espesor;
};
// otras implementaciones
bool Rectangulo :: seTraslapan(const Rectangulo &r) const {
    return ( !( r.x1 > x2 || r.x2 < x1 ||
                r.y1 > y2 || r.y2 < y1 ));
}

class RectanguloLleno : public Rectangulo {
public:
    // Otros métodos....
private:
    Color color;
}
```