

## Las preguntas con Opción múltiple Tienen UNA sola opción correcta.

**1er parcial: del 1 al 5**

**2do parcial del 6 al 10**

**Final 2-3-4-6-7-10**

**Instrucciones: copiar en la hoja el número y enunciado de la pregunta con la letra y la respuesta correcta.**

1. (1er parcial) Dada la clase producto, con una propiedad de instancia "descripción" (string) y el siguiente método en la misma clase.

```
Public void Mostrar()
```

```
{  
    Console.WriteLine("El la descripción del producto es: " +  
        producto.descripcion );  
}
```

- a. El método tiene errores de sintaxis
- b. Muestra la propiedad **descripcion** de un objeto producto
- c. No contiene errores de sintaxis pero mostraría la propiedad **descripcion** vacía
- d. Tiene que recibir un parámetro object para poder acceder a la propiedad **descripcion**
- e. a) y b) son correctas
- f. a) y d) son correctas

2. (1er parcial) ¿Qué diferencia hay entre una colección de tipo Generic con una No-Generic?

- a. No hay diferencia
- b. La Generic permite almacenar cualquier objeto, la otra no.
- c. Una es de tipo stack(pila) y la otra de tipo queue(cola).
- d. La primera es un array y la segunda una lista.
- e. Ninguna de las anteriores(justificar)

3. (1er parcial) Para que una propiedad pueda ser sobrescrita (sin obligación de hacerlo) y vista sólo por las clases derivadas

- a. Debe declararse como *Abstract virtual*
- b. Debe declararse como *private virtual*
- c. Debe declararse solo como *virtual*
- d. Una propiedad no puede tener más de un modificador
- e. Ninguna de las anteriores.(justificar)

4. (1er parcial) para sobrecargar un operador implícito la sintaxis correcta es:

- a. implicit override operador float(clase unObjeto){}
- b. public static implicit operador Clase(){}
- c. public static implicit operador int(clase unObjeto){}
- d. public implicit operador char(int a){}
- e. public static operador explicit Clase(int a){}

- f. Todas las anteriores
- g. Ninguna de las anteriores(justificar)

Universidad Tecnológica Nacional Facultad Regional Avellaneda									
									
Técnico Superior en Programación - Técnico Superior en Sistemas Informáticos									
Materia: Programación 2									
Apellido:				Fecha:					
Nombre:				Docente <sup>(2)</sup> :					
División:				Nota <sup>(2)</sup> :					
Legajo:				Firma <sup>(2)</sup> :					
Instancia <sup>(1)</sup> :	PP		RPP		SP		RSP		FIN

(1) Las instancias validas son: 1er Parcial (PP), Recuperatorio 1er Parcial (RPP), 2do Parcial (SP), Recuperatorio 2do Parcial (RSP), Final (FIN). Marque con una cruz.

(2) Campos a ser completados por el docente.

5. (1er parcial) Al utilizar una colección de tipo Cola

- a. Sólo se puede modificar un elemento
- b. Los objetos que almacena son de tipo QUEUE
- c. Al quitar un elemento, el elemento que se extrae es el último que se haya agregado
- d. Los elementos están encapsulados en propiedades de solo lectura
- e. Ninguna de las anteriores(justificar)
- f. b) y d) son correctas

6. (2do parcial) Dadas las clases A, B y C

- a. A puede heredar de B y de C
- b. Si A hereda de B, A hereda los constructores de B
- c. Si B hereda de A, y C hereda de A, entonces A debe ser una clase sellada.
- d. Si C hereda de B, y B hereda de A, entonces C hereda los miembros de A.
- e. Si B hereda de A, y C hereda de A, entonces A hereda los miembros de B
- f. Ninguna de las anteriores(justificar)

7. (2do parcial) Una interfaz:

- a. Los métodos NO pueden tener visibilidad
- b. Puede contener **solo** propiedades de solo lectura.
- c. Puede contener atributos de instancia
- d. Tiene constructores de instancia.
- e. No pueden tener atributos.
- f. Ninguna de las anteriores.
- g. Todas las anteriores
- h. a) y e) son correctas

8. (2do parcial) Una clase sellada que hereda de una clase abstracta

- a. Debe sobrescribir todos los métodos virtuales
- b. Una clase sellada no puede heredar de una clase abstracta
- c. Sólo puede implementar aquellos métodos abstractos y de instancia heredados que sean públicos o protegidos
- d. Solo debe implementar los métodos abstractos e implementación de la interface.
- e. Ninguna de las anteriores(justificar)

9. (2do parcial) ¿Cuándo NO puede crearse un método virtual?

- a. Cuando la clase está definida como *sealed*
- b. Cuando la clase pertenece a un namespace distinto de la clase *Program*
- c. Cuando la clase implementa una interfaz
- d. Cuando se trata de una clase hija
- e. Siempre se pueden crear.

10. (2do parcial) En las Excepciones :

- a. Los catch van de lo particular a lo general
- b. Por el finally pasa siempre.
- c. Cuando hay más de un try para el mismo catch, colocas la instrucción throw.
- d. Solo ocurre si no instanciamos las colecciones antes de serializar.
- e. El burbujeo es obligatorio en excepciones propias.
- f. a) y b) son correctas
- g. Ninguna de las anteriores(justificar)