

Pag. 1 Fecha: 30/OCT/2013 Hora: 11:27 hrs.

INSTITUTO TECNOLOGICO DE NUEVO LAREDO

ING. EN SISTEMAS COMPUTACIONALES

ING. BRUNO LOPEZ TAKEYAS

<<< P R E G U N T A S >>>

Unidad	Folio	Pregunta

POOC#	PROGRAMACION ORIENTADA A OBJETOS EN C#	
4 .	1	Caracteristica de POO que permite definir nuevas clases a partir de otras ya existentes a) Herencia b) Polimorfismo c) Encapsulamiento d) Sobre-escritura
4 .	2	Tipo de conexion entre clases que permite implementar relaciones de tipo "es un" a) Encapsulamiento b) Polimorfismo c) Herencia d) Abstraccion
4 .	3	Cuando se desea incorporar en una clase las variables, propiedades y metodos de otra clase, entonces se implementa ... a) Encapsulamiento b) Polimorfismo c) Herencia d) Abstraccion
4 .	4	Cuando se presentan relaciones de tipo "es un", entonces se implementa herencia a) Verdadero b) Falso
4 .	5	Cuando se desea incorporar en una clase las variables, propiedades y metodos de otra clase, entonces se implementa herencia a) Verdadero b) Falso
4 .	6	Identifique la(s) relacion(es) en el sig. codigo: class A : B { public A() { } } a) Herencia b) Polimorfismo c) Sobre-escritura d) Ninguna de las anteriores
4 .	7	La relacion de herencia entre clases es jerarquica a) Verdadero b) Falso
4 .	8	Es la habilidad que tienen los objetos para reaccionar de manera diferente ante los mismos mensajes a) Herencia b) Polimorfismo c) Encapsulamiento d) Sobre-escritura
4 .	9	Posibilidad de definir varias clases con funcionalidad diferente pero con metodos o propiedades definidos de manera identica a) Herencia b) Polimorfismo c) Encapsulamiento

- d) Sobre-escritura
- 4 . 10 El polimorfismo esta directamente relacionado con la sobrecarga y los metodos virtuales
a) Verdadero
b) Falso
- 4 . 11 Cuando un programa usa el mismo nombre en diferentes metodos con diferentes firmas, entonces se presenta ...
a) Herencia
b) Encapsulamiento
c) Sobrecarga
d) Sobre-escritura
- 4 . 12 El siguiente fragmento de codigo es un claro ejemplo de
Estudiante.Asistir();
Estudiante.Asistir(12);
a) Herencia
b) Encapsulamiento
c) Sobrecarga
d) Sobre-escritura
- 4 . 13 La sobrecarga representa diferentes maneras de realizar una misma accion
a) Verdadero
b) Falso
- 4 . 14 Para que se presente "sobre-escritura", debe haber relaciones de herencia
a) Verdadero
b) Falso
- 4 . 15 Cuando una clase derivada redefine los metodos y/o propiedades heredados, entonces se presenta ...
a) Abstraccion
b) Encapsulamiento
c) Sobrecarga
d) Sobre-escritura
- 4 . 16 Si una clase tiene metodos sobrecargados, entonces se presenta una modalidad de polimorfismo
a) Verdadero
b) Falso
- 4 . 17 Si una clase tiene propiedades sobrecargadas, entonces se presenta una modalidad de polimorfismo
a) Verdadero
b) Falso
- 4 . 18 Identifique el tipo de polimorfismo (si existe) en el siguiente fragmento de codigo:
class A {
 public void Usar() { ... }
 public void Usar(int x) { ... }
 public void Usar(string y) { ... }
}
a) Parametrico
b) Sobrecarga
c) Subtipo
d) No se presenta polimorfismo
- 4 . 19 Se puede presentar polimorfismo cuando clases diferentes e independientes tienen metodos y/o propiedades con el mismo nombre
a) Verdadero
b) Falso
- 4 . 20 Identifique el tipo de polimorfismo (si existe) en el siguiente fragmento de codigo:
class A {
 public void Usar() { ... }
}
class B {
 public void Usar() { ... } }
a) Parametrico
b) Sobrecarga
c) Subtipo
d) No se presenta polimorfismo
- 4 . 21 Se puede presentar polimorfismo cuando clases derivadas tienen metodos y/o propiedades sobre-escritos

- a) Verdadero
b) Falso
- 4 . 22 Identifique el tipo de polimorfismo (si existe) en el siguiente fragmento de código:

```
class A { public virtual void Usar() { ... } }  
class B : A { public override void Usar() { ... } }
```


a) Paramétrico
b) Sobrecarga
c) Subtipo
d) No se presenta polimorfismo
- 4 . 23 Cuando se sobrecargan métodos y/o propiedades de una clase, entonces se presenta polimorfismo de sobrecarga
a) Verdadero
b) Falso
- 4 . 24 Cuando se sobrecargan métodos y/o propiedades de una clase, entonces se presenta polimorfismo paramétrico
a) Verdadero
b) Falso
- 4 . 25 Los métodos sobrecargados del polimorfismo paramétrico se identifican por medio de su firma
a) Verdadero
b) Falso
- 4 . 26 En el polimorfismo paramétrico, los métodos y/o propiedades sobrecargados se comportan de manera diferente
a) Verdadero
b) Falso
- 4 . 27 En el polimorfismo paramétrico, los métodos y/o propiedades sobrecargados se ubican en la misma clase
a) Verdadero
b) Falso
- 4 . 28 En el polimorfismo paramétrico, los métodos y/o propiedades sobrecargados se ubican en clases diferentes
a) Verdadero
b) Falso
- 4 . 29 En el polimorfismo de sobrecarga, los métodos y/o propiedades que lo provocan se ubican en la misma clase
a) Verdadero
b) Falso
- 4 . 30 En el polimorfismo de sobrecarga, los métodos y/o propiedades que lo provocan se ubican en clases diferentes
a) Verdadero
b) Falso
- 4 . 31 En el polimorfismo de subtipo, se sobre-escriben algunos métodos heredados
a) Verdadero
b) Falso
- 4 . 32 Cuando una clase define varios métodos con el mismo nombre pero diferente firma, entonces se presenta polimorfismo ...
a) Paramétrico
b) Sobrecarga
c) Subtipo
d) No se presenta polimorfismo
- 4 . 33 La sobrecarga del constructor es un ejemplo de polimorfismo de tipo ...
a) Paramétrico
b) Sobrecarga
c) Subtipo
d) No se presenta polimorfismo
- 4 . 34 Cuando una clase define varios métodos con el mismo nombre pero diferente firma, entonces se presenta polimorfismo paramétrico
a) Verdadero

b) Falso
- 4 . 35 Cuando una clase define varios métodos con el mismo nombre pero diferente firma, entonces se presenta polimor-

- fismo de sobrecarga
a) Verdadero
b) Falso
- 4 . 36 Cuando una clase define varios metodos con el mismo nombre pero diferente firma, entonces se presenta polimorfismo de subtipo
a) Verdadero
b) Falso
- 4 . 37 Cuando una clase define varios metodos sobrecargados, entonces se presenta polimorfismo de sobrecarga
a) Verdadero
b) Falso
- 4 . 38 Cuando una clase define varios metodos sobrecargados, entonces se presenta polimorfismo parametrico
a) Verdadero
b) Falso
- 4 . 39 Cuando una clase define varios metodos sobrecargados, entonces se presenta polimorfismo de subtipo
a) Verdadero
b) Falso
- 4 . 40 La combinacion de herencia y metodos sobre-escritos producen un comportamiento polimorfico
a) Verdadero
b) Falso
- 4 . 41 La combinacion de herencia y metodos sobre-escritos producen un comportamiento polimorfico parametrico
a) Verdadero
b) Falso
- 4 . 42 La combinacion de herencia y metodos sobre-escritos producen un comportamiento polimorfico de sobrecarga
a) Verdadero
b) Falso
- 4 . 43 La combinacion de herencia y metodos sobre-escritos producen un comportamiento polimorfico de subtipo
a) Verdadero
b) Falso
- 4 . 44 La combinacion de herencia y metodos sobre-escritos producen polimorfismo ...
a) Parametrico
b) Sobrecarga
c) Subtipo
d) Ninguno de los anteriores
- 4 . 45 Si una clase base define un metodo abstracto, entonces obliga la implementacion de dicho metodo en sus clases derivadas
a) Verdadero
b) Falso
- 4 . 46 Un metodo abstracto de una clase base debe ser implementado en sus clases derivadas
a) Verdadero
b) Falso
- 4 . 47 Un metodo abstracto de una clase base no puede ser implementado en sus clases derivadas
a) Verdadero
b) Falso
- 4 . 48 Si diferentes clases tienen un metodo con el mismo nombre pero comportamiento diferente, entonces se produce un comportamiento polimorfico
a) Verdadero
b) Falso
- 4 . 49 Si diferentes clases tienen un metodo con el mismo nombre pero comportamiento diferente, entonces se produce un comportamiento polimorfico parametrico
a) Verdadero
b) Falso
- 4 . 50 Si diferentes clases tienen un metodo con el mismo nom-

- bre pero comportamiento diferente, entonces se produce un comportamiento polimorfico de sobrecarga
- a) Verdadero
b) Falso
- 4 . 51 Si diferentes clases tienen un metodo con el mismo nombre pero comportamiento diferente, entonces se produce un comportamiento polimorfico de subtipo
- a) Verdadero
b) Falso
- 4 . 52 Cuando una clase derivada redefine los metodos y/o propiedades heredados mediante la sobre-escritura, entonces se produce un comportamiento polimorfico
- a) Verdadero
b) Falso
- 4 . 53 Cuando una clase derivada redefine los metodos y/o propiedades heredados mediante la sobre-escritura, entonces se produce un comportamiento polimorfico parametrico
- a) Verdadero
b) Falso
- 4 . 54 Cuando una clase derivada redefine los metodos y/o propiedades heredados mediante la sobre-escritura, entonces se produce un comportamiento polimorfico de sobrecarga
- a) Verdadero
b) Falso
- 4 . 55 Cuando una clase derivada redefine los metodos y/o propiedades heredados mediante la sobre-escritura, entonces se produce un comportamiento polimorfico de subtipo
- a) Verdadero
b) Falso
- 4 . 56 Tipo de metodo definido en una clase base para ser sobreescrito en sus clases derivadas
- a) abstract
b) virtual
c) override
d) Todas las anteriores
- 4 . 57 Un metodo virtual puede ser sobre-escrito en una clase derivada
- a) Verdadero
b) Falso
- 4 . 58 Una clase base puede utilizar la palabra reservada "override" para definir un metodo y que pueda ser sobreescrito
- a) Verdadero
b) Falso
- 4 . 59 Una clase base utiliza la palabra reservada "virtual" para definir un metodo y que pueda ser sobre-escrito
- a) Verdadero
b) Falso
- 4 . 60 Una clase derivada utiliza la palabra reservada "override" para sobre-escribir un metodo
- a) Verdadero
b) Falso
- 4 . 61 Una clase derivada utiliza la palabra reservada "virtual" para sobre-escribir un metodo
- a) Verdadero
b) Falso
- 4 . 62 Un metodo virtual PUEDE ser sobre-escrito o utilizarse tal como esta
- a) Verdadero
b) Falso
- 4 . 63 Un metodo virtual DEBE ser sobre-escrito
- a) Verdadero
b) Falso
- 4 . 64 Una clase derivada SIEMPRE debe redefinir un metodo virtual

- tual
a) Verdadero
b) Falso
- 4 . 65 Una clase derivada PUEDE utilizar un metodo virtual tal como esta definido en la clase base
a) Verdadero
b) Falso
- 4 . 66 Se puede sobre-escribir un metodo abstracto de una clase base
a) Verdadero
b) Falso
- 4 . 67 Se puede sobre-escribir un metodo virtual de una clase base
a) Verdadero
b) Falso
- 4 . 68 Se puede sobre-escribir un metodo definido como "override"
a) Verdadero
b) Falso
- 4 . 69 Un metodo sobre-escrito debe mantener el mismo nivel de acceso que su respectivo metodo virtual
a) Verdadero
b) Falso
- 4 . 70 Es correcto el siguiente fragmento de codigo:

```
class A {
    public virtual void Calcular() { ... }
}
class B : A {
    public override void Calcular() { ... }
}
```

a) Verdadero
b) Falso
- 4 . 71 Para implementar polimorfismo es necesario que exista:
a) Agregacion
b) Composicion
c) Herencia
d) Una clase abstracta
- 4 . 72 Es correcto el siguiente fragmento de codigo:

```
class A {
    public override void Calcular() { ... }
}
class B : A {
    public virtual void Calcular() { ... }
}
```

a) Verdadero
b) Falso
- 4 . 73 Al definir un metodo "virtual" en una clase base, se le prepara para ser sobre-escrito
a) Verdadero
b) Falso
- 4 . 74 Para evitar la sobre-escritura de un metodo en futuras clases derivadas, entonces se declara como "override sealed"
a) Verdadero
b) Falso
- 4 . 75 Es correcto el siguiente fragmento de codigo:

```
class A { public virtual void Hacer() { ... } }
class B : A { public override sealed void Hacer() { ... } }

class C : B {
    public override void Hacer() { ... }
}
```

a) Verdadero
b) Falso
- 4 . 76 Si una clase base tiene definido un metodo virtual llamado Calcular(), Como se sobre-escribe dicho metodo en una clase derivada ?
a) public virtual void Calcular() { ... }

- b) `public override void Calcular() { ... }`
c) `public abstract void Calcular() { ... }`
d) No se puede sobre/escribir
- 4 . 77 En una clase base, Como se prepara un metodo llamado `Calcular()` para que pueda ser sobre-escrito en algunas clases derivadas ?
a) `public virtual void Calcular() { ... }`
b) `public override void Calcular() { ... }`
c) `public abstract void Calcular() { ... }`
d) Todas las anteriores
- 4 . 78 En una clase base, Como se prepara un metodo llamado `Calcular()` para que pueda ser sobre-escrito en algunas clases derivadas ?
a) `public virtual void Calcular() { ... }`
b) `public abstract void Calcular() { ... }`
c) `public override void Calcular() { ... }`
d) Todas las anteriores
- 4 . 79 Es correcto el siguiente fragmento de codigo:

```
class A { public virtual void Hacer() { ... } }  
class B : A { public override void Hacer() {  
    . . . }  
class C : B {  
    public override void Hacer() { ... }  
}
```


a) Verdadero
b) Falso
- 4 . 80 Es correcto el siguiente fragmento de codigo:

```
class A { public virtual void Hacer() { ... } }  
class B : A { public override void Hacer() {  
    . . . }  
class C : B {  
    public override sealed void Hacer() { ... }  
}
```


a) Verdadero
b) Falso
- 4 . 81 Es correcto el siguiente fragmento de codigo:

```
class A { public void Hacer() { ... } }  
class B : A {  
    public override void Hacer() { ... }  
}
```


a) Verdadero
b) Falso
- 4 . 82 Es correcto el siguiente fragmento de codigo:

```
class A { public abstract void Hacer();  
}  
class B : A {  
    public override void Hacer() { ... }  
}
```


a) Verdadero
b) Falso
- 4 . 83 Es correcto el siguiente fragmento de codigo:

```
class A { public abstract void Hacer();  
}  
class B : A { public override void Hacer() { ... }  
}  
class C : B { public override void Hacer() { ... }  
}
```


a) Verdadero
b) Falso
- 4 . 84 Se puede reemplazar un metodo heredado por otro nuevo, diferente, con el mismo nombre y la misma firma e introducirlo en la jerarquia de clases
a) Verdadero
b) Falso
- 4 . 85 Si se reemplaza un metodo heredado por uno nuevo, entonces se "sustituye" el codigo del nuevo metodo por el de la clase base
a) Verdadero
b) Falso
- 4 . 86 Se puede ocultar un metodo heredado e introducir uno nuevo a la jerarquia de clases

- a) Verdadero
b) Falso
- 4 . 87 Para ocultar un metodo heredado e introducir uno nuevo a la jerarquia de clases, es necesario que clase base lo defina como "virtual"
a) Verdadero
b) Falso
- 4 . 88 Para ocultar un metodo heredado e introducir uno nuevo a la jerarquia de clases, es necesario que clase base lo defina como "abstract"
a) Verdadero
b) Falso
- 4 . 89 Para ocultar un metodo heredado e introducir uno nuevo a la jerarquia de clases, es necesario que clase base lo defina como "override"
a) Verdadero
b) Falso
- 4 . 90 Si tiene el siguiente fragmento de codigo:

```
class A {  
    public void Pintar() { ... }  
}
```


Como se oculta dicho metodo en una clase B derivada de la clase A (class B : A) ?

a) public override void Pintar() { ... }
b) public abstract void Pintar() { ... }
c) public new void Pintar() { ... }
d) Ninguna de las anteriores
- 4 . 91 En una clase derivada, Para ocultar y sustituir un metodo heredado e introducirlo a la jerarquia de clases se le antepone la palabra "new"
a) Verdadero
b) Falso
- 4 . 92 En una clase derivada, Para ocultar y sustituir un metodo heredado e introducirlo a la jerarquia de clases se le antepone la palabra "virtual"
a) Verdadero
b) Falso
- 4 . 93 En una clase derivada, Para ocultar y sustituir un metodo heredado e introducirlo a la jerarquia de clases se le antepone la palabra "abstract"
a) Verdadero
b) Falso
- 4 . 94 En una clase derivada, Para ocultar y sustituir un metodo heredado e introducirlo a la jerarquia de clases se le antepone la palabra "sealed"
a) Verdadero
b) Falso
- 4 . 95 Se puede sobrecargar el destructor de una clase
a) Verdadero
b) Falso
- 4 . 96 La diferencia entre sobrecarga y polimorfismo es que la primera se presenta en tiempo de compilacion y el polimorfismo se presenta en tiempo de ejecucion
a) Verdadero
b) Falso
- 4 . 97 La diferencia entre sobrecarga y polimorfismo es que la primera se presenta en tiempo de ejecucion y el polimorfismo se presenta en tiempo de compilacion
a) Verdadero
b) Falso
- 4 . 98 Las clases abstractas obligan la herencia
a) Verdadero
b) Falso
- 4 . 99 Las interfaces obligan la herencia
a) Verdadero
b) Falso
- 4 . 100 Una clase abstracta es lo mismo que una interfase

- a) Verdadero
b) Falso
- 4 . 101 Una clase abstracta puede ser instanciada
a) Verdadero
b) Falso
- 4 . 102 Una interfase puede ser instanciada; es decir, se pueden crear objetos de ella
a) Verdadero
b) Falso
- 4 . 103 Una clase abstracta puede ser heredada
a) Verdadero
b) Falso
- 4 . 104 Una interfase puede ser heredada
a) Verdadero
b) Falso
- 4 . 105 Se pueden crear objetos de una clase abstracta
a) Verdadero
b) Falso
- 4 . 106 Se pueden crear objetos de una interfase
a) Verdadero
b) Falso
- 4 . 107 Una clase abstracta puede heredar de otras clases
a) Verdadero
b) Falso
- 4 . 108 Una clase abstracta puede heredar de interfaces
a) Verdadero
b) Falso
- 4 . 109 Una clase abstracta puede contener atributos (datos)
a) Verdadero
b) Falso
- 4 . 110 Una clase abstracta puede contener implementacion de metodos y propiedades
a) Verdadero
b) Falso
- 4 . 111 Los miembros abstractos de una clase deben ser implementados en las clases derivadas
a) Verdadero
b) Falso
- 4 . 112 En UML, una clase abstracta se representa con el nombre en itálicas
a) Verdadero
b) Falso
- 4 . 113 En UML, una clase abstracta se representa con el nombre subrayado
a) Verdadero
b) Falso
- 4 . 114 En UML, una clase abstracta se representa con el nombre repintado (bold)
a) Verdadero
b) Falso
- 4 . 115 En UML, un miembro abstracto de una clase se representa en itálicas
a) Verdadero
b) Falso
- 4 . 116 En UML, un miembro abstracto de una clase se representa subrayado
a) Verdadero
b) Falso
- 4 . 117 En UML, un miembro abstracto de una clase se representa repintado (bold)
a) Verdadero
b) Falso

- 4 . 118 Un metodo abstracto de una clase contiene el codigo de su implementacion
a) Verdadero
b) Falso
- 4 . 119 Una clase abstracta solamente puede tener miembros abstractos
a) Verdadero
b) Falso
- 4 . 120 Una clase abstracta puede tener combinaciones de metodos abstractos y metodos normales
a) Verdadero
b) Falso
- 4 . 121 Como se declara una clase abstracta llamada Persona
a) class abstract Persona
b) abstract class Persona
c) class Persona abstract
d) class Abstract Persona
- 4 . 122 Se pueden definir miembros abstractos dentro de una clase abstracta
a) Verdadero
b) Falso
- 4 . 123 Se pueden definir miembros abstractos dentro de una clase no abstracta
a) Verdadero
b) Falso
- 4 . 124 En una clase derivada, los metodos abstractos heredados se definen como "override"
a) Verdadero
b) Falso
- 4 . 125 En una clase derivada, los metodos abstractos heredados se definen como "virtual"
a) Verdadero
b) Falso
- 4 . 126 Una interfase contiene la implementacion de sus metodos y/o propiedades
a) Verdadero
b) Falso
- 4 . 127 Una interfase contiene atributos (datos)
a) Verdadero
b) Falso
- 4 . 128 Una interfase puede heredar de otras clases
a) Verdadero
b) Falso
- 4 . 129 Una interfase puede heredar de otras interfaces
a) Verdadero
b) Falso
- 4 . 130 La diferencia entre una clase abstracta y una interfase es que esta ultima no contiene datos
a) Verdadero
b) Falso
- 4 . 131 Es correcto el siguiente fragmento de codigo:
abstract class Vehiculo
{
 ...
}
Vehiculo Avion = new Vehiculo();
a) Verdadero
b) Falso
- 4 . 132 Es correcto el siguiente fragmento de codigo:
abstract class Vehiculo
{
 public abstract void Encender();
}
class Auto : Vehiculo
{
 public override void Encender() { ... }
}
a) Verdadero
b) Falso

- 4 . 133 Es correcto el siguiente fragmento de codigo:
abstract class Vehiculo
{
 public virtual void Encender();
}
class Auto : Vehiculo
{ public override void Encender() { ... } }
a) Verdadero
b) Falso
- 4 . 134 Es correcto el siguiente fragmento de codigo:
class Vehiculo
{
 public abstract void Encender();
}
class Auto : Vehiculo
{ public override void Encender() { ... } }
a) Verdadero
b) Falso
- 4 . 135 Se puede sobre-escribir un metodo definido "abstract"
a) Verdadero
b) Falso
- 4 . 136 Se puede sobre-escribir un metodo definido "virtual"
a) Verdadero
b) Falso
- 4 . 137 Se puede sobre-escribir un metodo definido "override"
a) Verdadero
b) Falso
- 4 . 138 Se puede definir un metodo abstracto en una clase abstracta
a) Verdadero
b) Falso
- 4 . 139 Se puede definir un metodo abstracto en una clase no abstracta
a) Verdadero
b) Falso
- 4 . 140 Es correcto el siguiente fragmento de codigo:
abstract class Vehiculo
{
 public abstract void Encender() { . . . }
}
class Auto : Vehiculo
{ public override void Encender() { ... } }
a) Verdadero
b) Falso
- 4 . 141 Es correcto el siguiente fragmento de codigo:
class Vehiculo { public abstract Encender(); }
class Auto : Vehiculo { public override Encender()
 { . . . }
}
class AutoCompacto : Auto
{ public override void Encender() { ... } }
a) Verdadero
b) Falso
- 4 . 142 Es correcto el siguiente fragmento de codigo:
class Vehiculo { public override Encender() {...} }
class Auto : Vehiculo { public override Encender()
 { . . . }
}
class AutoCompacto : Auto
{ public override void Encender() { ... } }
a) Verdadero
b) Falso
- 4 . 143 Es correcto el siguiente fragmento de codigo:
class Vehiculo { public abstract Encender() {...} }
class Auto : Vehiculo { public override Encender()
 { . . . }
}
class AutoCompacto : Auto
{ public override void Encender() { ... } }

- a) Verdadero
b) Falso
- 4 . 144 Como se define un metodo de una clase para que pueda ser sobre-escrito ?
a) virtual
b) abstract
c) override
d) Todas las anteriores
- 4 . 145 Es correcto el siguiente fragmento de codigo:

```
abstract class Vehiculo { public abstract Encender();  
                        public virtual Apagar() { ... }  
}  
class Auto : Vehiculo {  
    public override void Encender() { ... }  
    public override void Apagar() { ... } }
```


a) Verdadero
b) Falso
- 4 . 146 Una clase que declara metodos virtuales ...
a) nunca es abstracta
b) puede ser abstracta
c) siempre es abstracta
d) no seria una clase, sino una interfase
- 4 . 147 Considere el siguiente codigo:

```
abstract class A {  
    public double Consultar();  
}
```


Considere una clase B que hereda de A. Como se sobre-escribe el metodo Consultar() en la clase derivada ?
a) public override double Consultar() { ... }
b) public double Consultar() { ... }
c) public override double consultar() { ... }
d) Ninguna de las anteriores
- 4 . 148 Un metodo abstracto puede contener implementacion; es decir, puede tener codigo
a) Verdadero
b) Falso
- 4 . 149 Un metodo abstracto obliga la sobre-escritura
a) Verdadero
b) Falso
- 4 . 150 En una clase, Como se puede declarar un metodo llamado Calcular() para que pueda ser sobre-escrito en las clases derivadas ?
a) public virtual void Calcular() { ... }
b) public abstract void Calcular();
c) public override void Calcular() { ... }
d) Todas las anteriores
- 4 . 151 En una clase, Como se puede declarar un metodo llamado Calcular() para que pueda ser sobre-escrito en las clases derivadas ?
a) public void Calcular();
b) public void abstract Calcular() { . . . }

c) public void override Calcular();
d) Ninguna de las anteriores
- 4 . 152 Una clase abstracta puede contener datos (atributos)
a) Verdadero
b) Falso
- 4 . 153 Explique la diferencia entre un metodo virtual y un metodo abstracto
- 4 . 154 Explique la diferencia entre una clase normal y una clase abstracta
- 4 . 155 Explique la diferencia entre sobrecarga y sobre-escritura
- 4 . 156 Como se define un metodo de una clase base para que pueda ser redefinido en una clase derivada
a) virtual
b) override
c) abstract
d) Todas las anteriores

- 4 . 157 Como se define un metodo de una clase derivada para sobrescribirlo
 a) virtual
 b) override
 c) abstract
 d) Todas las anteriores
- 4 . 158 Definicion de un metodo en una clase base que no contiene implementacion
 a) virtual
 b) abstract
 c) override
 d) sealed
- 4 . 159 Definicion de un metodo en una clase para impedir la sobrescritura en clases derivadas sucesivas
 a) virtual
 b) abstract
 c) override
 d) sealed
- 4 . 160 Como se declara una clase llamada Auto que implementa el metodo de la interfase IEquatable
 a) class Auto : IEquatable
 b) class Auto : IEquatable<>
 c) class Auto : IEquatable<Auto>
 d) Ninguna de las anteriores
- 4 . 161 Como se declara una clase llamada Auto que implementa el metodo de la interfase IComparable
 a) class Auto : IComparable<>
 b) class Auto : IComparable<Auto>
 c) class Auto : IComparable (Auto)
 d) class Auto : IComparable Auto
- 4 . 162 Nombre de la interfase que sirve para determinar si dos objetos son iguales
 a) IEqualable
 b) IEnumerable
 c) IEquatable
 d) Ninguna de las anteriores
- 4 . 163 Nombre de la interfase que sirve para determinar si un objeto es menor que otro
 a) IComparable
 b) IEnumerable
 c) IEquatable
 d) Ninguna de las anteriores
- 4 . 164 Metodo de una interfaz del Framework que determina si un objeto especificado es igual al objeto actual
 a) CompareTo()
 b) ==
 c) =
 d) Equals()
- 4 . 165 Metodo de una interfaz del Framework que compara la instancia actual con otro objeto del mismo tipo para determinar el orden relativo de los objetos comparados
 a) CompareTo()
 b) <, >, <=, >=, ==, !=
 c) Compare()
 d) Equals()
- 4 . 166 Si se declara ...

```
class Auto:IEquatable<Auto> {
    public string Placas { get; set; }
}
```

 Como se declara el metodo para comparar si dos autos son iguales ?
 a) public void Equals(Auto x)
 b) public int Equals(Auto x)
 c) public bool Equals(Auto x)
 d) private bool Equals(Auto x)
- 4 . 167 Si se declara ...

```
class Auto:IComparable<Auto> {
    public string Placas { get; set; }
}
```

 Como se declara el metodo para comparar el orden relativo

vo de dos autos ?

- a) public void CompareTo(Auto x)
- b) public int CompareTo(Auto x)
- c) public bool CompareTo(Auto x)
- d) private int CompareTo(Auto x)

- 4 . 168 Si se declara ...

```
class Persona : IEquatable<Persona> {
    public int Edad { get; set; }
    public bool Equals(Persona x) { ... }
}
```

 Como se implementa el metodo para comparar si dos personas tienen la misma edad ?
 a) return(if(this.Edad == x.Edad));
 b) return(this.Edad = x.Edad);
 c) return(this.Edad == x.Edad);
 d) return(Edad == x.Edad);
- 4 . 169 Si se declara ...

```
class Persona : IComparable<Persona> {
    public int Edad { get; set; }
    public int CompareTo(Persona x) { ... }
}
```

 Como se implementa el metodo para determinar si una persona es mayor que otra ?
 a) if(this.Edad > x.Edad) return 1; else return 0;
 b) return(this.Edad > x.Edad);
 c) if(this.Edad >= x.Edad) return 1; else return 0;
 d) return(Edad > x.Edad);
- 4 . 170 Se puede implementar tanto el metodo Equals() de la interfase IEquatable como el metodo CompareTo() de la interfase IComparable en una misma clase
 a) Verdadero
 b) Falso
- 4 . 171 Como se define una clase llamada Auto que desea implementar el metodo Equals() de la interfase IEquatable y el metodo CompareTo() de la interfase IComparable ?
 a) class Auto : IEquatable, IComparable
 b) class Auto : IEquatable<Auto>, IComparable<Auto>
 c) class Auto : IEquatable<Auto> IComparable<Auto>
 d) No se pueden implementar dos interfaces en una clase
- 4 . 172 Si una clase implementa de varias interfaces, entonces se considera herencia multiple
 a) Verdadero
 b) Falso
- 4 . 173 La siguiente interfase:

```
interface Auto {
    string placas;
    string MostrarDatos();
}
```

 Es incorrecta porque ...
 a) El nombre de la interfase no comienza con "I"
 b) No se ha colocado abstract o virtual en sus elementos
 c) Se ha definido un atributo dentro de la interfase
 d) Se ha omitido el modificador de acceso "public"
- 4 . 175 La sobrecarga del constructor es un ejemplo de polimorfismo
 a) Verdadero
 b) Falso
- 4 . 176 El destructor de una clase puede provocar comportamiento polimorfo al objeto correspondiente
 a) Verdadero
 b) Falso
- 4 . 177 Cuales son las similitudes y diferencias entre una clase abstracta y una interfase?
- 4 . 178 Bajo que circunstancias recomienda utilizar una interfase?
 Y una clase abstracta?
- 4 . 179 Cuando una clase hereda de varias interfaces, se considera herencia multiple?
 Por que?

=====

TOTAL: 178 PREGUNTAS