**Quiz 2**

**Integrantes:** Laura Natalia Gonzalez Mora, Sergio Esteban Rendon Umbarila,Pedro Antonio Perez Sibaja,Diego Raul Guerrero Anaya

**1. Responda Verdadero o Falso para las siguientes enunciados.  
a.** Si una subclase de una superclase abstracta no implementa todos los métodos abstractos, la subclase debe ser definida como abstracta.  
 *Verdadero*.  
**b.** Un método abstracto puede ser estático.  
 *Falso*.  
**c**. No se pueden crear instancias de una clase abstracta usando el operador new, pero una clase abstracta puede ser usada como tipo de dato.  
 *Verdadero*.  
**d.** Una clase abstracta puede ser extendida.  
 *Verdadero*.  
e. Una subclase de una superclase no abstracta no puede ser abstracta.  
 *Verdadero*.  
**f.** Una subclase no puede sobreescribir un método concreto en una superclase para definirlo como abstracto.  
 Todas las clases son subclases (heredan de Object), por tanto es FALSO. La afirmación no implica que la primera clase a la que se hacer referencia (subclase) sea una subclase de la segunda clase a la que se hacer referencia (superclase)..  
**g.** Un método privado en la superclase puede ser sobreescrito en la subclase.  
 Falso  
**h.** Un método estático puede ser heredado, Sin embargo, un método estático no puede ser sobreescrito (Override).  
 *Los métodos estáticos no se heredan y por eso no se pueden sobrescribir.*  
**i.** Son equivalente las siguientes especificaciones?  
 *No son equivalente por el static de la variable y el abstract del método.*  
**j.** Una interface puede tener métodos estáticos  
 Falso. Los métodos son abstractos.  
**k.** Una interface puede extender uno o más interfaces  
 *Verdadero. Una clase interface puede ser implementada por cualquier número de clases o heredar de varias interfaces.*  
**l.** Una interface puede extender una clase abstracta.   
 *Falso. Un interfaz no puede extender clases. Aunque un interfaz puede extender múltiples interfaces.*  
**m.** Una clase abstracta puede extender una interface.  
 *Una clase abstracta puede IMPLEMENTAR una interface, no extender.*  **2.**

1. salida :

1

0

1. salida:

0

1. Excepción runtimeException

b.

**3.**

**A.**

1. Salida: **A`s no-arg constructor is invoked.**
2. El error està en que el constructor de la clase A no puede aplicarse a los tipos dados. El constructor de B requiere un parámetro derivado del constructor del padre(A), pero no hay ningún parámetro en el constructor de B. Por pantalla se imprime esto: **Excepción en el subproceso "main" java.lang.RuntimeException, o lo que es Exception runtimeException.**

**B.**

a) **F**

b) **F**

c) **F**

**Nota:** Lo que sucede es que ninguna de las dos clases tienen métodos por lo cual no se pueden ejecutar.