

# Examen segundo parcial - Teoría

**Fecha de entrega** 1 de sep en 12:30      **Puntos** 30      **Preguntas** 2

**Disponible** 1 de sep en 11:30 - 1 de sep en 12:30 casi 1 hora

**Límite de tiempo** 30 minutos

Este examen fue bloqueado en 1 de sep en 12:30.

## Historial de intentos

	Intento	Hora	Puntaje
MÁS RECIENTE	<a href="#">Intento 1</a>	30 minutos	16 de 30

❗ Las respuestas correctas están ocultas.

Puntaje para este examen: **16** de 30

Entregado el 1 de sep en 12:08

Este intento tuvo una duración de 30 minutos.

### Pregunta 1

6 / 15 pts

Observe el código a continuación. El código producirá 5 errores de compilación. Para cada error escriba el número de línea donde se encuentra y explique brevemente el motivo del mismo.

```
1 package modelo;
2 public interface InterfaceA{
3     void metodo1();
4     default void metodo2(){
5         System.out.println("P00 ");
6     }
7 }
```

```
1 package modelo;
2 public interface InterfaceB extends InterfaceA{
3     void metodo1b();
4     void metodo2b();
5 }
```

```
1 package modelo;
2 public abstract class MiClase implements InterfaceB{
3     public void mostrar(){
4         System.out.println("Salida");
5     }
6     public void metodo1b(){
7         System.out.println("Java ");
8     }
9     public void metodo2b(){
10        System.out.println("ESPOL ");
11    }
12 }
```

```
1 package modelo;
2 public class MiHija extends MiClase{
3     void metodo2(){
4         System.out.println("POO 2023 ");
5     }
6     public static void main(String args[]){
7         InterfaceA.metodo2();
8         InterfaceA obj1 = new MiHija();
9         obj1.mostrar();
10        InterfaceB obj2 = new MiClase();
11        obj2.metodo2b();
12    }
13 }
```

Su respuesta:

En la clase Mihija de ley tiene que sobreEscribir al metodo 1 de la interfaceA

No podemos crear una instancia de una clase abstracta como en la linea 10

Linea 8 no se puede ya que es una interface

## Pregunta 2

10 / 15 pts

1. Cree una excepción de tipo checked llamada **TestException** con un constructor que recibe el mensaje de lo ocurrido.
2. El siguiente código producirá 2 errores de compilación. Para cada error escriba el número de línea donde se encuentra y **explique brevemente** cómo resolverlo.

```
1 v class ClaseA{
2 v     void cambiar() throws RuntimeException{
3         throw new TestException("Err");
4     }
5 v     void mostrar(){
6         System.out.println("Test");
7     }
8 }
9
10 v class ClaseB extends ClaseA{
11     @Override
12 v     void mostrar() throws Exception{
13         throw new Exception();
14     }
15 }
```

Su respuesta:

```
public class TestException extends Exception() {
```

```
    public TestException(String msg){
```

```
        super(msg);
```

```
    }
```

```
}
```

en la línea 2 ya que debe lanzar el **throws TestException**,  
**RuntimeException**

en la línea 13 no puede sbreescribirlo porque no es

Puntaje del examen: **16** de 30