1. ¿Qué significa sobrecargar un método o constructor?  
   Es cuando dos o más métodos en una clase comparten el mismo nombre pero con diferentes parámetros, número, tipo u orden, es decir se cambia la firma del método.
2. ¿Qué debe cambiar para que la sobrecarga de un método o constructor sea válida?  
   Tiene que cambiar los parámetros, ya sea número, tipo u orden.
3. ¿La sobrecarga se resuelve en tiempo de ejecución o en tiempo de compilación? ¿Cómo se distingue a que sobrecarga llamar?  
   Se resuelve en tiempo de …. y se distingue por los parámetros que se le pasan al método o constructor.
4. ¿Se tiene en cuenta el nombre o identificador de los parámetros de entrada para una sobrecarga?  
   No, solo se tiene en cuenta el tipo, número, u orden.
5. ¿Se tiene en cuenta el modificador de visibilidad para una sobrecarga?
6. ¿Los métodos pueden tener el mismo nombre que otros elementos de una misma clase?  
   No, no pueden. Pueden tener el mismo nombre que otro método siempre y cuando sea una sobrecarga.
7. ¿Mencione dos razones por las cuales se sobrecargar los métodos?  
   Si hay métodos similares que requieren parámetros difernetes, añadir funcionalidad al código existente.
8. ¿Los métodos estáticos pueden ser sobrecargados?  
   Si, si se puede.
9. ¿Agregar el modificador “static” sin cambiar los parámetros de entrada es una sobrecarga válida?  
   No.
10. ¿Agregar un modificador “out” o “ref” en la firma del método sin cambiar nada más es una sobrecarga válida?  
    Si.
11. ¿Cambiar el tipo de retorno?  
    No.
12. Si tenemos distinta sobrecargas de un método, ¿Cómo podemos reutilizar código?  
    Llamando a alguno de los métodos ya sobrecargados.
13. ¿Se pueden sobrecargar los constructores estáticos?  
    No, ya que no recibe parámetros ni tiene modificadores de acceso, es llamado automáticamente en tiempo de ejecución la primera vez que se inicializa un campo estático
14. ¿Se puede llamar a un constructor estático con el operador “this()”?  
    No.
15. ¿Se puede llamar a constructores de otras clases con el operador “this()”?  
    No.
16. ¿Se puede sobrecargar un constructor privado?  
    Si.
17. ¿Qué es un operador? ¿En qué se diferencian un operador unario y un binario? Ejemplos.  
    Los operadores unarios realizan una acción con un solo operando. Los binarios realizan acciones con dos operando.  
    Ejemplo de unario: ++, --, ¡, &.  
    Ejemplo de binario: +, -, \*, /.
18. ¿Qué varía en la sintaxis de la sobrecarga de operadores unarios y binarios?  
    La sobrecarga de operadores unarios puede no recibir un argumento, salvo que este sea global, lo cual debe recibir uno de tipo de clase.  
    Los binarios deben recibir un argumento de tipo de clase, si es global debe recibir dos, y uno de ellos debe ser del tipo de clase.
19. ¿Se pueden sobrecargar los operadores de operación y asignación (+=, -=, etc)?  
    Cuando se sobrecarga un operador binario, también se sobrecarga implícitamente el operador de asignación correspondiente. Por lo que no puede sobrecargar de forma explicita un operador de asignación.
20. ¿Cuál es la diferencia entre un operador de conversión implicito y uno explicito?  
    Los operadores de conversión permiten hacer compatibles tipos que antes no lo eran.  
    Se declaran igual cambiando la palabra explicit o implicit y son estáticos.
21. ¿Los operadores de casteo “(T)x” no se pueden sobrecargar. ¿Cuál es la alternativa?  
    Definir nuevos operadores de conversión.
22. ¿caster, convertir, parsear?  
    Casting es cuando tomas una variable de tipo y la cambias a otro sin alterar el valor de la variable.   
    Converting similar a casting con la diferencia que puede ser entre 2 tipos no compatibles.  
    Parsear Cuando interpretas y convertís un string a un tipo diferente leyendo su contenido.