### ****1. What is stored in the static heap, stack, and dynamic heap?****

* **Static Memory (Method Area)**: Stores class-level data, including method definitions, static variables, and metadata of classes.
* **Stack Memory**: Stores local variables and method call frames. Each method call creates a new stack frame that is removed once the method execution is completed.
* **Heap Memory (Dynamic Heap)**: Stores objects and instance variables. When new is used to create an object, memory is allocated in the heap.

### ****2. What are objects in the program?****

Objects are instances of classes. If the program involves:

Guitar obj1 = new Guitar();

Guitar obj2 = new Guitar("SN123", 1500, "Fender", "Stratocaster", "Mahogany", "Spruce");

Then, obj1 and obj2 are objects of the Guitar class, stored in the heap.

### ****3. What is the state of obj1, obj2?****

* obj1: Created using the default constructor, meaning all fields contain empty values ("", 0, or false for booleans).
* obj2: Created using the parameterized constructor, meaning it has specific values assigned.

### ****4. Do you access all fields of obj1 in the class**** Tester.java****? Why?****

Yes, but only if Tester.java properly calls the getter methods or directly accesses public fields (if any). If the fields are private, they can only be accessed through getters and setters.

### ****5. What is the current object when the program runs the line**** obj2.createSound();****?****

At that moment, obj2 is the current object because the method createSound() is called on it. Inside createSound(), this refers to obj2.

### ****6. In the**** main ****method, can you use the keyword**** this ****to access all fields of obj2? Why?****

No, this cannot be used in a static method like main(). The main() method belongs to the class, not an instance. Since this refers to the current object instance, it cannot be used in static contexts.