Abstraction: Using a name to represent a more complicated program fragment/idea.

Binding: when we attach a name t a concept.

Binding times:  
1. Language design time  
2. Language implementation time  
3. Program writing time  
4. Compile time  
5. Link time  
6. Load time  
7. Run time

Binding lifetime: How long the binding exists

Object lifetime: How long the object exists

1. Stack allocated: allocated on the stack frame, object is on the stack as long as the frame is on the stack  
2. Heap allocated: allocated from heap, it exists until explicitly deallocated  
3. Static: the object is allocated by the compiler or operating system

Scoping: when/where in a program a binding exists and is visible

1. static scoping: the scope of a binding can be determined by reading the code.   
 Examples:   
 🡪 local variables where the scope of the binding is the block where it is declared  
 🡪 global variables where scope is entire program  
 🡪 scope can be entire module  
2. Dynamic scoping: used in original lisp and scripting languages. The binding of a variable is the closest definition on the call stack.