1. What is a stack frame and what kind of information is stored in a stack frame?
Activation record for a method. Stack frame is all of the information needed to run a JVM method. This includes the constant pool, local variables, heap, etc.

2. What is the lifetime scheme of information stored in stack frames?
The lifetime scheme of information stored in stack frames is "Invocation lifetime", or until it is returned by some method.

3. In what way does the JVM provide "stack frames" for method calls? That is, how does the jasmin programmer view the "stack frames" of methods?
Stack frames, based on their relation to the programmer, are relavent by the parameters passed

4. Where do calling methods put actual parameters for method calls?
With an invokevirtual/invokespecial instruction, Arguments are popped from the stack along with their address's, a new stack frame is created and the arguements get pushed onto that new environment's stack.

5. Where does the JVM store the actual parameters for the called method to access? That is, where does the callee access the incoming parameters? Assume the method is a static method.
The callee accesses the incoming parameters from local 0.

6. Now explain where the JVM puts actual parameters for virtual method calls and how they are accessed by the callee. That is, explain how the parameter passing for virtual methods is different from the parameter passing for static methods that you explained above.
The first parameter is an object and goes in local 0.

7. How do you return a value from a method call?
Use a typed return: I, F, Ljava/lang/String;

8. Can a method access the stack frame of its caller? If so, how? If not, then why not?
You can't get to the stack frame of the caller.