

### 1. Static vs. dynamic

- (a) Define the terms “static” and “dynamic”, with respect to program properties.
- (b) Give two examples of program properties constructs or language features which may have static and dynamic variants.
- (c) For one of your chosen examples, discuss the benefits and tradeoffs between the static and dynamic variant.

### 2. Virtual machine subsystems

Below is a table of the four classes of virtual machines and a set of VM subsystems. Fill out the table based on what subsystems would be generally necessary for that class of VM.

VM subsystem	Hypervisor	Emulator	Bytecode VM	Language VM
code loader				
typechecker				
interpreter				
JIT compiler				
object model				
debugger				
exception handling				
garbage collector				
disassembler				
text parser				

### 3. Bytecode virtual machines vs. language virtual machines.

- (a) Give three examples of bytecode VMs and language VMs.
- (b) What is the primary difference between language VMs and bytecode VMs?
- (c) Discuss two advantages that bytecode and language VMs have over the other.

### 4. Binary formats

- (a) Give three examples of code formats for binaries and briefly state their primary use case.
- (b) List three properties that would be considered ideal for a code format.
- (c) List three properties that would be considered ideal for a code format.
- (d) For each one of the properties you listed above, discuss potential drawbacks that might occur if a code format designer were to focus only on that property.

### 5. Bytecode formats

Give one advantage and one disadvantage each for stack machines (SM) and register machines (RM) as a bytecode format.