

Jerry Huang

Period 2

APCS

Kuszmaul

Vocabulary 7

1. (invariant - representation) A representation invariant is a computer programming construct consisting of a set of invariant properties that remain uncompromised regardless of the state of the object.
2. (invariant - algorithm) A special algorithm can be written to make an invariant work properly and more efficiently.
3. (invariant - compile) The theory of optimizing compilers relies heavily on invariants.
4. (invariant - interpret) Abstract interpretation tools can compute simple invariants of given imperative computer programs.
5. (invariant - virtual) An invariant can be used in a game designed through virtual programming.
6. (invariant - indirection) We can access the position of the invariant through indirection.
7. (representation - algorithm) An algorithm can be a representation of an invariant.
8. (representation - compile) An intermediate representation can help combine compiling and interpreting.
9. (representation - interpret) Most interpreted languages use an intermediate representation, which combines compiling and interpreting.
10. (representation - virtual) In the Java virtual machine, false is represented by integer zero and true by any non-zero integer.
11. (representation - indirection) An intermediate representation is a data structure that is constructed from input data to a program while indirection refers to accessing a variable through a pointer.
12. (algorithm - compile) An algorithm in the program is interpreted after the code is compiled.
13. (algorithm - interpret) An interpreter cannot detect logical errors in an algorithm.

14. (algorithm - virtual) Virtual machines can be made more efficient if the algorithm used to deploy it is improved.
15. (algorithm - indirection) Indirection is a fundamental aspect of algorithms.
16. (compile - interpret) When a program is being compiled, the interpreter converts the code to machine language in order to execute it.
17. (compile - virtual) The Java Virtual Machine interprets compiled Java binary code for a computer's processor so that it can perform a Java program's instructions.
18. (compile - indirection) One can distinguish indirections exist only prior to compile time and indirections that continue to exist in the runtime.
19. (interpret - virtual) The Java Virtual Machine can interpret code in order to perform instructions.
20. (interpret - indirection) In strongly typed interpreted languages with dynamic data types, most variable references require a level of indirection.
21. (virtual - indirection) A virtual machine is an emulation of a particular computer system while indirection is the ability to reference something using a name.