

1.

(a)

B: <w,2>,<j,3>,<k,3>,<h,1>,<i,1>

A: <x, 8>,<y, 8>,<i, 9>,<j, 9>,<h, 1>

Main: <a,15>,<b,15>,<h,1>,<i,1>

(b) B: <w,2>,<j,3>,<k,3>,<a,15>,<b,15>,<h,1>,<i,1>

(c) A: <x, 8>,<y, 8>,<i, 9>,<j, 9>,<a,15>,<b,15>,<h, 1>

(d) B: <w,2>,<j,3>,<k,3>,<x, 8>,<y, 8>,<i,9>,<a,15>,<b,15>,<h,1>

2. C++'s namespace mechanism allows programmers to declare a new namespace in which new identifiers such as variables and functions are declared. The major benefit is to group related identifiers into the same namespace and multiple namespaces can have an identifier with the same name but with different implementations. Client code can open a namespace and import all identifiers declared in the namespace (in a sense similar to Java's import mechanism).

3. (1) No. If a variable is not allocated in memory, it does not exist, and hence cannot be used and invisible to any scope.

(2) Yes. An instance variable (declared at the class scope) can be hidden by a local variable. As another example, in a static Java method, all non-static Java fields are invisible even though those fields may be allocated in some object.