1. Suppose someone designed a stack abstract data type in which the function top returned an access path (or pointer) rather than returning a copy of the top element. This is not a true data abstraction. Why? Give an example that illustrates the problem.

Data abstraction is the reduction of a body of data to a simplified representation of the whole. In other words, a means to reduce something to only the essential data. If a pointer was returned instead of the real value in the stack, what happens if the program was trying to manipulate data in that class? Would the methods in the class return the correct result if an operation on the data requires the true value of the data itself? What happens on increment or decrement? The implementation of such would open to numerous problems.

1. Describe three ways a client can reference a name from a namespace in C++.

a.

namespace someNamespace{

int i, j;

}

- This for local use, simply groups the i and j names in a namespace called, “someNamespace”

b.

someNamespace::i

someNamespace::j

- This version references the i and j from the predefined “someNamespace”

c.

using namespace std;

* Note: this one imports all the namespace names from std for use of the class

1. What are the arguments for and against the Objective-C design that method access cannot be restricted?

Pro

You can ask an object if it has a particular method / selector

You can ask for the signature of the method

Basically, the above relates to good introspection.

Objective C has dynamic runtime which means you can add methods to existing classes and change method implementations

Con

No private methods

1. What are the arguments for and against the C++ policy on inlining of methods?

Pro

Faster, no need to push and pop objects on / off the stack

You can put a function definition in a header file

Con

Encapsulation problems

Larger header file

The compiler may or may not inline the functions

1. What are the advantages and disadvantages of the ability to change objects in Ruby?

Pro

Methods can also be removed from a class.

New members can be added at any time to a class [because classes are dynamic].

Predefined classes can be extended

In other words, it’s a flexible language

Con

Poor readability

Hard to trace or follow program