1. **Java Interface Dispatch**

Consider the following two interface dispatch schemes, as described in the assigned paper by Alpern, et al.:

(i) Directly Indexed ITables (section 4.1, para. 2)

(ii) Interface Method Tables with 5 entries (IMT-5) (section 5)

(a) Characterize a set of Java programs that cause method (i) to use large amounts of space for data structures, but that method (ii) handles well. Explain why.

(b) Conversely, characterize a set of Java programs that cause method (ii) to use large amounts of space and time for conflict resolution stubs, but that method (i) handles well. Explain why.

(c) According to the authors, how did the performance of these two schemes compare on real programs?

2. **.NET Generics**

Consider the assigned paper by Kennedy and Syme.

(a) Describe a significant difference, having to do with types, between JVM bytecode and the original (non-generic) .NET IL.

(b) The authors claim that there are advantages to instantiating generics in the context of a JIT compiler. What are some of these advantages?

3. **Scheme Interpretation**

There are hardly any type error checks in Michaux’s scheme interpreter.

(a) What is his reason for omitting them?

(b) Add appropriate type error checking code to the subtraction primitive procedure in scheme.c. Submit your revised version of scheme.c, or better yet, just the results of a diff with the original version.

**How to submit your homework.**

Submit the homework by mail to apt@cs.pdx.edu with subject line “CS577 HW2.” Put the answers to all questions into a single plain text file and attach it to your mail.