QB2Solution

[A , 1 points ]Fill-in:

[A1] Run-time interrogation becomes necessary because of the Liskov Substitution Principle,

and can be achieved in the Eiffel language using an \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ attempt (?=).

(assignment)

[B, 4 points ] Multiple Choice:

[B1] Inheritance sets up a(n) \_\_\_\_\_\_\_\_\_\_\_\_\_\_ hierarchy. (a) (is-a)

(a) is-a (b)has-a

[B2] Aggregation sets up a(n) \_\_\_\_\_\_\_\_\_\_\_\_\_\_ hierarchy. (b) (has-a)

(a) is-a (b)has-a

[B3] Code reuse can be achieved using \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_. (c)

(a)inheritance (b) genericity (c)both inheritance and genericity

[B4] The problem of repeated inheritance can be solved using \_\_\_\_\_\_\_\_\_\_\_\_\_\_ classes (c)

(a)mixin classes (b)role-playing classes (c)both mixin and role-playing classes.

[C, 4 points ] Short Answer

[C1] State the Liskov Substitution Principle.

You can assign to a variable of an ancestor type an object or variable of a descendant type.

[C2] State the Open-Closed Principle

A class should be closed so that it can be used but should be open for extension (via inheritance)

[C3] State the Law of Inversion

If functions exchange too much data, put the functions in the data

(explains transition from structured to Object-Oriented Design).

[C4] State the Single Responsibility Principle.

(A class should have a well-defined single purpose. This facilitates code reuse).

[D , 6 points] True/False

[D1] Inheritance can always be replaced by aggregation+delegation. (T)

[D2] Aggregation+delegation can always be replaced by Inheritance. (F)

[D3] Structured control structures are preferred over programming with GOTOs. (T)

[D4] Conditional code should be replaced by a combination of inheritance,

polymorphism and dynamic binding.(T)

[D5] C++ supports garbage collection. (F)

[D6] Java supports multiple inheritance. (F)