2009 November 23

# General Design
# Example Test Questions

**1.**

Why is design necessary? What is its purpose?

**2.**

**A** What makes an expression referentially transparent?

**B** What does it mean for a function to have side effects? How does a side effect of a function affect the referential transparency of an expression?

**3.**

Explain what the Uniform Access Principle is. Using an example illustrate the benefits of following the Uniform Access Principle in Eiffel.

**4.**

Explain what is the direct mapping rule?

**5.**

The *Open-Closed Principle* states that a good module structure should be both closed and open. This double requirement looks like a dilemma, which is solved in object oriented software construction. Explain the principle and describe how it is solved in OOSC.

**6.**

Explain what is an expanded type and why expanded types are needed.

**7.**

Describe and explain the "Explicit Interfaces Rule".

**8.**

Explain what the "Explicit Interfaces Rule" is and the context in which it is used.

**9.**

What is the self-documentation principle? How does Eiffel support this principle?

**10.**

Explain the four principles of public program design.

**11.**

What is the single choice principle?

**12.**

Among design rules to support modularity, briefly describe the following rules:

small interface rule

few interface rule

explicit interface rule

**13.**

Explain what is meant by seamlessness in the context of software design.

**14.**

Give definitions of the following with respect to software design.

**Error Defect Fault**

**15.**

What are active data structures? When would they be used? Why are they used?

**16.**

What is the distinction between design and program?

**17.**

We have a group of related objects such as the following that need to be shared among several classes, say A, B, C. Describe, in the context of object-oriented design, **two** different methods, other than the *Singleton* pattern, of sharing these global objects.

```
Pi: REAL is 3.141592
Mean_radius_of_earth: REAL is 6371.01  -- mean radius of earth in kilometer
```

**Part A:** Describe how **Method 1** works

What is the major advantage of using this method?
What is the major disadvantage of using this method?

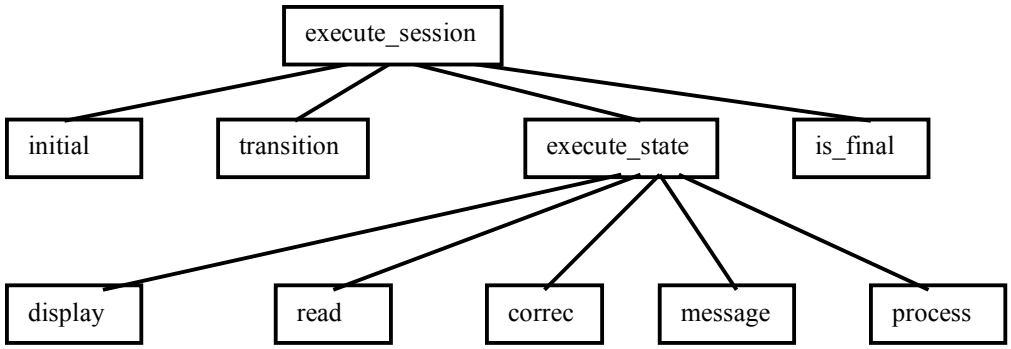
**Part B:** Describe how **Method 2** works

What is the major advantage of using this method?
What is the major disadvantage of using this method?

**18.**

Recall that a Multi-Panel Interactive System is a general type of interactive system in which users interact with a set of panels, filling in blanks in each panel and moving to other panels by making choices. Each session goes through a number of states, starting with an initial state and ending with a final one. Each state corresponds to an interactive panel.

Also recall that the top-down functional approach to this problem gave the following solution:




Where execute_session and execute_state are defined as follows:

```
Execute_session is
  -- Execute a complete session
  local state, next : INTEGER
  do
    state := initial  -- start in initial state
    repeat
          -- next is a var parameter in execute
      execute_state (state,  next )
      state := transition ( state, next )
    until is_final ( state )
  end
```

```
execute_state ( in s : INTEGER , out c : INTEGER ) is
  -- c contains the user's choice for next state
  local a : ANSWER  ;  ok :BOOLEAN
  do
    repeat
       display (s )    -- display panel for state s
       read ( s , a )   -- get user answer in a
        ok := correct ( s , a )
    until ok end
    process ( s , a )
    c := next_choice ( a )   -- get user choice for panel
  end
```

**A** It is suggested that various modules, given by this solution, are tightly coupled with each other and with the choice of application. Explain why.

**B** Explain why this solution violates the Single-Choice Principle.

**C** Does this solution provide reusable parts? Explain your answer.

**19.**

Describe each of the following object oriented design principles: abstraction, encapsulation, modularity.

**20.**

Modular design issues deal with information hiding and independence. Explain what is meant by these terms

**21.**

Briefly describe the following terms in the context of modular design and complete the complete the relevant sentence:

Cohesion: a module's property indicating ???
Coupling: a module's property indicating ???
It is desirable to have software modules with ??? coupling and ??? cohesion

**22.**

It is desired for a software construction methodology that supports decomposition and composition. Briefly describe each term:

Decomposition Composition

**23.**

t has been suggested that decomposability and composablity are sometimes contradictory. Explain why.