2009 November 23

BON
Example Test Questions

1.

For each of the following requirements, identify candidate classes (with appropriate names) and define their relationships using BON. Associations and aggregations must be labeled. If no operations are specified, then do not provide any class interface details and use the short ellipse notation for classes. If operations are specified, then give appropriate interfaces and class invariants but do not give any contracts. Briefly explain each diagram in English.

1. A father or a mother may have a son or daughter. A son or daughter may have a brother or a sister.

2. A vehicle may be rented on a certain date. The rental will be at a specific rate and will be for a specific driver. Insurance is optional.

3. A conference session may be a tutorial session, paper session or demo session. Each session involves one or more presentations. Paper presentations are reviewed, while tutorial presentations are not.

4. A person can be a teacher or a student. A teacher can teach many courses. A student can enroll in many courses. A teacher can mentor students but a student only has one mentor. State in which class(es) you would insert a class invariant that a teacher may not be a student in a class that they teach. Give, as mathematically as possible the class invariant(s).

5. A folder in a personal computer can contain folders, text files, binary files, and pictures. Folders, files and pictures can be opened and closed. Folders, files and pictures can be inserted and removed from a folder.

6. A system is made up of clusters, classes and relationships. Clusters may contain other classes and clusters. Relationships have a source and a target, and connect clusters or classes. A relationship is either of type association, inheritance or aggregation. The latter two relationships cannot connect a class or cluster to itself.

7. The Acme widget company depends upon its employees, including its owner, its contract employees and its full time employees. All employees have a unique employee id.

8. Instructors at a university can be either tenure stream or part-time. Courses can have one or more sections. Courses with more than one section must have tenure stream instructors teaching one or more sections. Exactly one of the tenure stream instructors in a multi-section course is a course director. The instructor of a single section course is also the course director. Tenure stream instructors teach at most 3 courses. Part time instructors teach at most 6 courses. Give a static BON diagram modeling these ideas. Include appropriate class invariants. Associations (has a) and aggregations must have labels. Briefly describe your model.

9. An employer is a person who hires many employees. An employee is a person who works on several projects. An employee works for only one employer.

10. Political parties, such as the Conservative party, the New Democratic Party and the Liberal party have both supporters and voters.

11. Draw a BON static diagram for a *graphical document editor* that supports grouping. Grouping allows a collection of graphical concepts (e.g. ellipses, squares, etc.) to be grouped and moved/edited as a whole. A document is composed of several sheets. Each sheet contains drawing objects, including text, geometric shapes, and groups. A group is a set of drawing objects that must contain at least two drawing objects. Each drawing object is a member of at most one group. Geometrical objects include circles, ellipses, rectangles, lines and squares.

Your diagram should only consist of the short form for classes (ellipses only) and relationships. Associations and aggregations must have labels. Do not show any class interface details, except class invariants, if you think they are necessary. Most classes should not need invariants.

12. A food company is in the business of buying food from producers and selling it wholesale to its customers. The company would like to design a system to keep track of its inventory and its contacts. The inventory consists of a list of food items some of which are fresh food with an expiry date attached to them. The company maintains a list of contacts of both producers and customers. Each contact has an address. Each customer has a credit rating and each producer has a catalogue of food items it offers. Customers are not producers, nor are producers customers.

Include appropriate attributes and types, in suitable classes, so that your system can provide the following information.

1. A list of customers with a good credit rating.
2. A list of producers offering a specific food item.
3. All fresh food items that will expire on certain date.

The design can be at a high level of abstraction, using generic classes such as SET and LIST. For the purpose of this question we are only interested in overall architecture rather than implementation and efficiency.

13. A dental office has a number of patients. A patient is either male or female. Each patient has a set of teeth. The office would like to keep track of each patient's information including address and date of birth as well as information regarding his/her complete set of teeth. For each tooth a history is kept which is simply a note indicating the condition of that tooth. In addition, the office would like to keep a record of all operations performed on a tooth together with the date of the operation. An operation could be a root canal or an extraction. If a tooth is extracted then no other operation may be performed on that tooth. A tooth may have at most one root canal operation.

Your system must be able to query the system for the following information.

1. A list of patients with a root canal operation.
2. The number of teeth extractions performed in a month.
3. All operations performed on female patients within certain date.

The design should be at a high level of abstraction, using deferred classes such as SET and LIST. You may assume that lower level classes such as DATE and ADDRESS exist. For the purpose of this question, we are only interested in overall architecture rather than implementation and efficiency. Identify constraints that are class invariants and indicate the classes that contain those invariants.

14. A coffee shop offers several types of hot drinks including coffee, hot chocolate, and tea. It also has a sitting area with a minimum of 5 seats. The price of tea is less than the price of coffee and coffee is cheaper than hot chocolate. A coffee shop has at least one owner.

2.

Describe, in development order, the nine development steps for the BON process to analyze and design a system – include static and dynamic aspects. Include, in the earliest step that an artifact is produced, a brief description of the artifact. Take time to organize your answer. It must fit within the allotted pages. Note instructions 4 and 6.

3.

A Draw a diagram and describe the components of the BON static model.

B Draw a diagram and describe the components of the BON dynamic model.

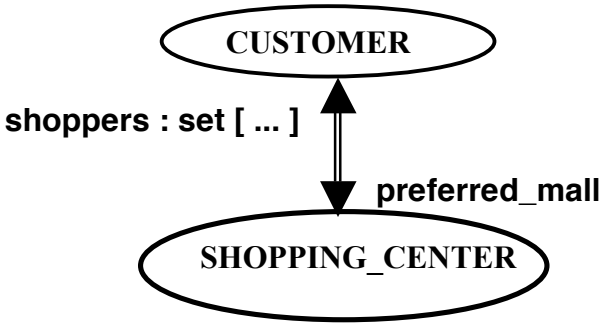
4.

A Explain and clarify the difference between an association relationship and an aggregate relationship.

B When might you use aggregation but not association?

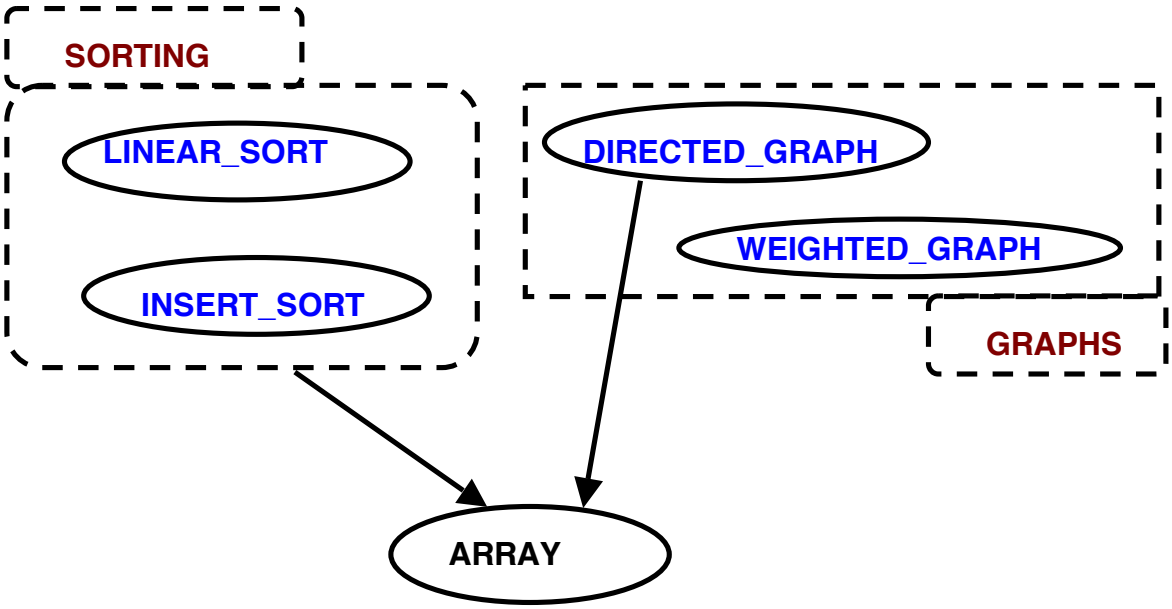
5.

Explain the following BON diagram



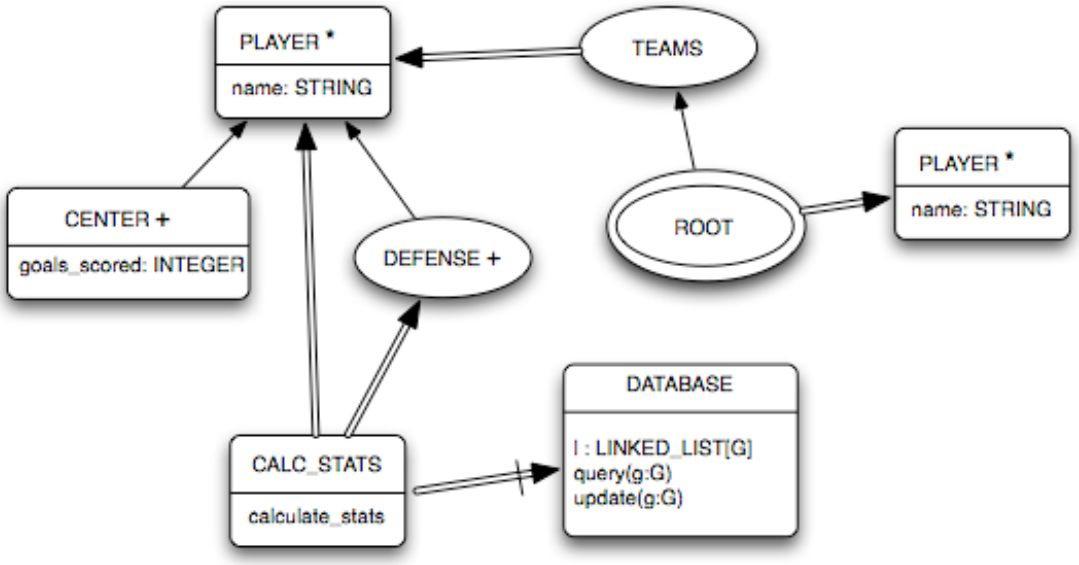
6.

Explain the following BON diagram



7.

You are a software engineer at a company owned by Montgomery Burns, the most evil man in Parry Sound. One day, Mr. Burns, who thinks he knows something about software design, comes to you with a proposed design for a web-based hockey pool manager. This system is intended to maintain professional hockey team statistics and provide a central database so that everyone in the pool – mostly lazy employees – can access the players' statistics. Mr. Burn's proposed design is as follows.



Mention **five** things that you think are wrong or flawed with this design, and explain why you think they are flawed. **NOTE:**

You don't need to know anything about hockey in order to understand this design and its flaws. The flaws are due to poor OO design practice.

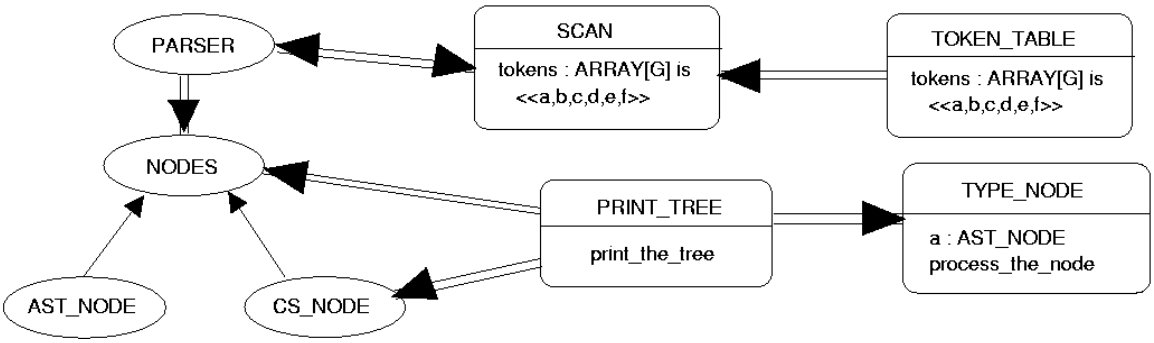
A label missing from a client-supplier relationship is not considered a flaw.

There are *at least* five flaws. Only the first five you write down will be considered.

Mr. Burns will fire you for criticizing his design.

8.

A student who slept through most of the lectures in COSC 3311 this term decided to design a compiler. They proposed the following design.



Mention five things that you think are wrong or flawed with this design, and explain why you think they are flawed. **NOTE:**

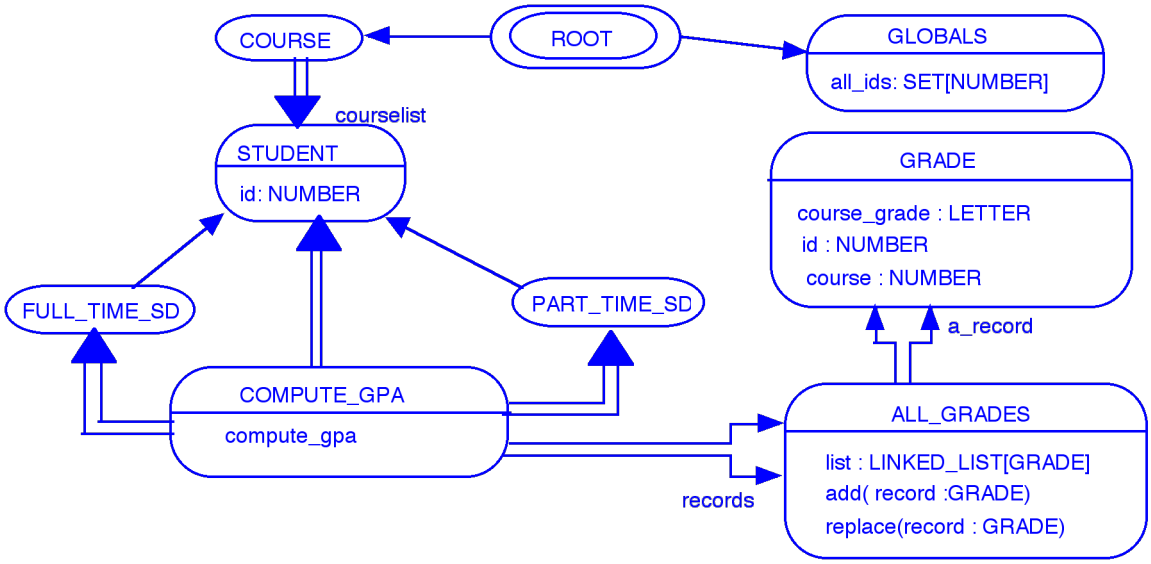
You don't need to know anything about compilers in order to understand this design and its flaws. The flaws are due to poor OO design practice.

<<a,b,c,d,e,g>> is a constant array consisting of six characters.

There are *at least* five flaws.

9.

The Magnificent Software Company proposed the following design as a grade database, and to compute student gpa's for prerequisite checking and course grade profiles.



Describe 5 and no more than 5 things that you think are wrong or flawed (an error) with this design. Clearly number from 1 to 5 each error you list. Explain clearly why you believe each is an error. A label missing from a client-supplier relation is not considered an error.

10.

Consider the Eiffel classes provided in the following table. In the space provided below give a BON diagram representing the relationships between these classes. Use compressed graphical notation for classes. Make sure to indicate whether a class is deferred or effective and to label your diagram as needed.

Class *EARTH* ***inherit*** *PLANET* ***feature*** *the_planet*: *EARTH* ***is*** ***once*** *Result* := *Current* ***end*** ***end*** -- class *EARTH*	***class*** *EARTH_ACCESSOR* ***inherit*** *PLANET_ACCESSOR* ***rename*** *planet* ***as*** *the_earth* ***end*** ***feature*** *the_earth*: *EARTH* ***is*** ***once*** ***create*** *Result.make* (*"..."*) ***end*** ***end*** -- class *EARTH_ACCESSOR*
Deferred class *PLANET* ***feature*** {*NONE*} *the_planet*: *PLANET* ***is*** ***deferred*** ***end*** ***end*** -- class *PLANET*	***deferred class*** *PLANET_ACCESSOR* ***feature*** {*NONE*} *planet*: *PLANET* ***is*** ***deferred*** ***end*** ***end*** -- class *PLANET_ACCESSOR*