Family name _____Solution_____

Given name(s) _____

Student number _____

York University
Faculty Science & Engineering / Faculty of Arts
Department of Computer Science & Engineering

Class Test 2

AK/AS/SC – CSE 3311 3.0
Software Design

2012 March 12

Instructions

1. This is a closed book examination. No examination aids are permitted – no calculators, telephones, etc.
2. All questions are to be attempted.
3. Answer each question in the space provided.
4. Each question is evaluated on the York University letter grade scale A+, A, ... , D, E, F.
5. Where descriptive answers are requested, use complete sentences and paragraphs
6. All programming and mathematical work is to be annotated; include comments explaining what you are doing.
7. Wherever appropriate, use diagrams to enhance your answers.
8. If a question is ambiguous or unclear then please write your assumptions and proceed to answer the question.
9. You may write in pencil but class tests written in pencil are not re-evaluated.
10. The test time is approximately 75 minutes.

Ques	Max	Mark
1	9	_____
2	9	_____
3	9	_____
4	9	_____
5	9	_____
Total		_____
Letter grade		_____

Question 1

A Explain when one can use design by contract?

In your own words explain the diagram in slides 20 & 21 in the set on Design by Contract with respect to when to use design by contract. See next comment. Can have design by contract even if programming environment does not directly support it as Eiffel does. It is a mind set and method of working and documenting programs.

B Explain when one cannot use design by contract?

In your own words and explain the diagram in slides 20 & 21 in the set on Design by Contract with respect to when not to use design by contract. Good place for a diagram. Need more than saying outside the system, as you have to explain what is meant by within the system and outside the system – which system? Within York you and I are within the same system but we cannot use design by contract to enforce communications between us. Why? Because we cannot be reliably relied upon to follow a contract, such as correctly filling in a form or following instructions as a computer can be. There is no need to introduce notions of ethical and moral behaviour.

0

C Describe the benefits and obligations of the client and supplier when using design by contract.

Expand on Design by Contract slides 12, 16 & 17.

Need to clearly specify what obligations and benefits for each of client and supplier with respect to pre and post conditions. Not sufficient to just state that pre- and post-conditions are the benefit and obligations; i.e. include why and when are they benefits and obligations?

Not enough to say strong or weak conditions. Need to mention explicitly that strength deals with the amount of work involved and relate to who the the responsibility to do the work and who benefit from the work done.

Question 2

In mathematical notation, fill in the indicated contract assertions. **Do not** use agents.

Each study group has a name and a list of its members. Each student has a name and a list of the names of the study groups in which they want to be members.

```
class STUDY_GROUP                class STUDENT
  name    : STRING                 name     : STRING
  members : SET[STUDENT]           in_group : SET[STRING]
end                              end
```

```
class STUDENT_ASSOCIATION
  study_groups : SET[STUDY_GROUP]    -- All the study groups in the association
  members      : SET[STUDENT]        -- All the students in the association

make(students : SET[STUDENT] , group_names: SET[STRING])
  -- Creates an association with the study groups according to the preferences of the students.
  -- students – the students making the association
  -- group_names – the list of names of all the study groups in the association
```

require

-- The names of the study groups that students want to be in are in *group_names*.

$\forall\ s \in students \bullet s.in\_group \subseteq group\_names$

alternate $\forall\ s \in students \bullet (\forall\ sg \in s.in\_group \bullet sg \in group\_names)$

-- Between 3 and 5 students want to be in each study group.

$\forall\ name \in group\_names \bullet 3 \leq \#\{\ s \in students \mid name \in s.in\_group \bullet s\ \} \leq 5$

ensure

-- There are no members of the association other than the students making the association and all
-- the students are members of the association.

$members = students$

alternate $\forall\ m \in members\ ;\ s \in students \bullet m \in students\ \hat{}\ s \in members$

-- The study groups in the association are precisely those with names in *group_names*.

$group\_names = \{\ sg \in study\_groups \bullet sg.name\ \}$

alternate $(\forall\ sg \in study\_groups \bullet sg.name \in group\_names) \wedge \#study\_groups = \#group\_names$

Question 2 – continued

```
number_of_groups(size : INTEGER) : INTEGER
```
-- Returns the number of study groups that have *size* members

ensure

Result = #{sg ∈ study_groups | #sg.members = size • sg}

invariant

-- The members of every study group are all the people who want to be in that study group.

∀ sg ∈ study_groups • sg.members = { m ∈ members | sg.name ∈ m.in_group}
alternate ∀ sg ∈ study_group ; m ∈ members • m ∈ sg.members ⇔ sg.name ∈ m.in_group

There are no arrays in the program text, as a consequence, array notation cannot be used in the answers, no such features

end – STUDENT_ASSOCIATION

Question 3

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 kilometers
```

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

Slides Global Objects 23-3 and 23-4 and Section 18.2 of the textbook.

Describe inheriting the class CONSTANT. Require a BON diagram

Nothing to do with patterns, the only pattern that does the job is the Singleton pattern and it was excluded.

What is the major advantage of using this method?

Have direct access to the variables, no need to introduce other variables.

What is the major disadvantage of using this method?

The client class is in an inappropriate is_a relationship with the class CONSTANT but is unlikely to be used in a polymorphic way. Have multiple copies of the constants.

Question 3 – continued

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

Define the class CONSTANT and use it as a supplier. Require a BON diagram

Nothing to do with patterns, the only pattern that does the job is the Singleton pattern and it was excluded.

What is the major advantage of using this method?

The client class is not in a false is_a relationship.

What is the major disadvantage of using this method?

Have to use indirection to access the constants. Have multiple copies of the constants.

Question 4

A Consider the following program text.

```
class THING feature
   the_thing: THING
      once
         Result := Current
      end
invariant
   only_thing: the_thing = Current
end
```

```
class BETTER
   inherit THING
end
```

```
class WORSE
   inherit THING
end
```

Assuming the class invariant is checked, does the execution of the following statements lead to any assertion violation in Eiffel? Justify your answer.

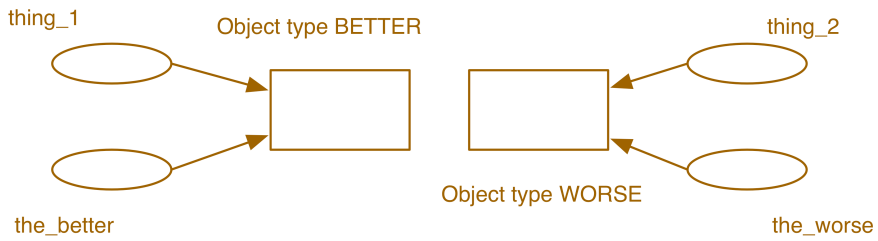
```
thing_1, thing_2 : THING
the_worse: WORSE
the_better: BETTER

create the_worse
create the_better
thing_1 := the_better.the_thing
thing_2 := the_worse.the_thing
```

Yes! When "create the_better" is executed, the class invariant in THING will become false (violated). Current in THING refers to the_better whereas the_thing points to the_worse.

B Assume class BETTER, from part A, is modified as defined below. Draw a diagram of the objects created as a result of execution of the set of statements in Part A. Assume assertions are turned on.

```
class BETTER
   inherit THING redefine thing end
feature
   thing: BETTER once Result := Current end
end
```

Typo here should be the_thing, would be good to mention it, but people understood it was the_thing



thing_1 has the same value as the_better and thing_2 has the same value as the_worse.

Question 5

A Using Bon, give the generic static diagram of the Singleton pattern and include relevant interface features.

Slide Singleton-13

B Give the scenario and object communication (dynamic model) diagram for the Singleton pattern.

Slide Singleton-3

Question 5 – continued

C Give and describe the memory diagram for the singleton pattern. Describe the relationship between the scenario and what is created in memory.

Give and describe slide Singleton-4

For remarking you need to write a note stating clearly and exactly where you believe your grade should be increased or decreased. Remember that the grade is a qualitative one. You need to explain, if you think your grade should go up, why you believe the quality of your answer is good (B) and not competent (C+), or, if you think the grade should go down, why you believe the quality of your answer is very good (B+) and not excellent (A).

The entire test will be reevaluated. Your grade may go up, it may stay the same, or it may go down. I will look over the entire test and see if the grades good, excellent, minimal, etc are applicable to the work as a whole (see the web page on grading in the course) independent of the points assigned to the parts.