Student Name:

Student Number:

Midterm Examination
COMP 304B 2003: Object-oriented Design

Examiner: Prof. Hans Vangheluwe Monday, March 3rd, 2003

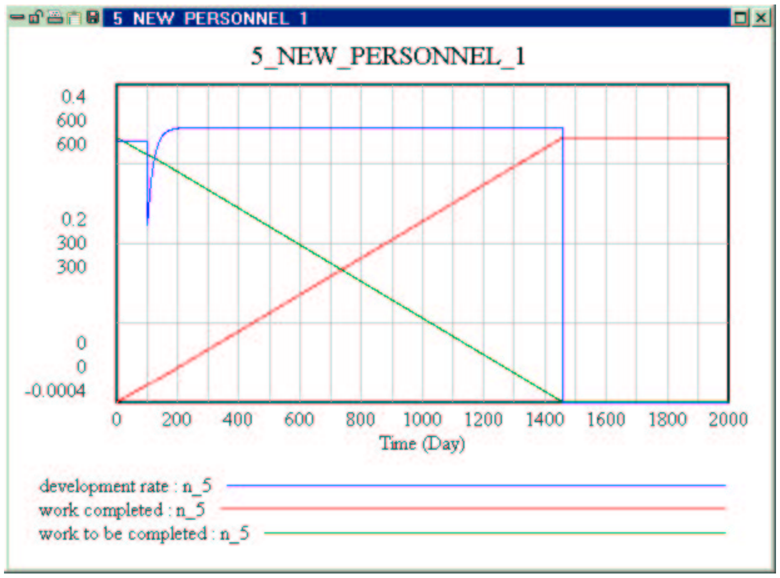
Invigilators: Jean-Sébastien Bolduc, Spencer Borland 14:30 – 16:00

INSTRUCTIONS:

1. Answer all questions directly on the examination paper.
2. No aids of whatever type are permitted.
3. The exam has 10 questions on 10 pages (including cover page).
4. Attempt all questions: partial marks are given for incomplete but correct answers.
5. Numbers between brackets [] denote the weight of each question. The exam is out of a total of 40 points.
6. The midterm counts for 15 % of the total COMP 304 grade.
7. Use the back of the last page as scrap (it will be ignored during grading). The rear of the other pages may be used as extra space to answer questions.

Good luck !

(1) [1]



What does the above figure teach us about software development team size ?

(2) [3]

- What are the characteristics of good unit tests ? Note: the answer is *not* the different types of tests.

- Suppose we want to test a class `StackOfIntegers` which implements the wellknown stack data-structure. The class has the following public methods with the obvious meanings:
  - `push(x:Integer)`
  - `pop() -> Integer`
  - `topOfStack() -> Integer`

- `isEmpty() -> Boolean`

For the above, give a small example of each different type of test which needs to be done (use pseudo-code to give the essence of the test only).

(3) [11]

Give the 9 generally accepted features of Object-Oriented systems with a short explanation for each. For each of the features, draw (an example of) the UML notation, if applicable.

1.

2.

3.

4.

5.

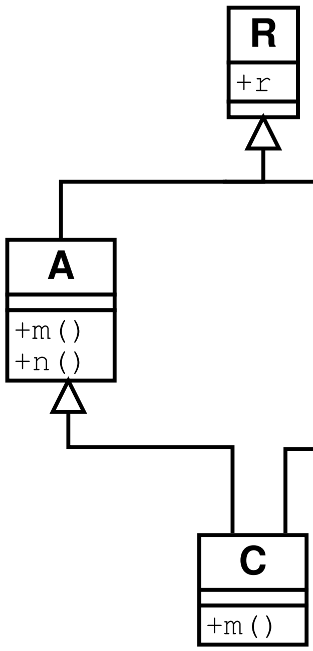
6.

7.

8.

9.

(4) [5]



1. Referring to the above figure, explain the problems with repeated and multiple inheritance. What are the alternative interpretations/implementations ?

2. `a, b, c` are instances of `A, B, C` respectively. If the multiple inheritance semantics of Python is used for `class C(A,B)`, which class' method (circle the appropriate one) will be invoked with

   (a) `a.m()` `A B C none`

   (b) `a.n()` `A B C none`

   (c) `a.o()` `A B C none`

(d) `b.m()` A B C none

(e) `b.n()` A B C none

(f) `b.o()` A B C none

(g) `c.m()` A B C none

(h) `c.n()` A B C none

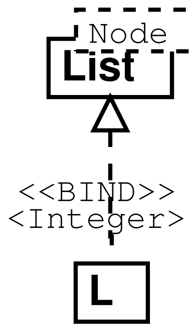
(i) `c.o()` A B C none

3. What does one call the relationship between method `m()` in classes `A` and `C` ?

4. Is the attribute `r` in `R` a part of (the state of) instances of `C` ? YES NO.

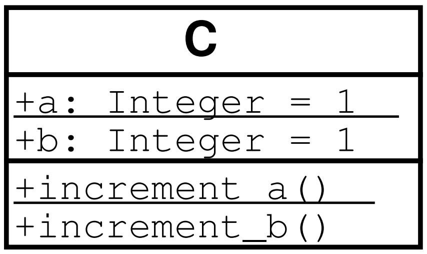
5. If the attribute `a` in `R` were *private*, would `a` be part of instances of `C` ? YES NO.

(5) [1]



Explain the above figure.

(6) [1]



For the class definition above, and after sequential execution of (using Python syntax)

```
c1=C()
c2=C()
```

```
c1.increment_a()
c1.increment_b()
c2.increment_a()
c2.increment_b()
```

give the values of

- `c1.a`
- `c1.b`
- `c2.a`
- `c2.b`

Note how `a` and `increment_a()` are underlined !

(7) [7]

Draw the *class diagram* for the following: A Building can be specialized in many ways. In this design, we only show two: Highrise and TownHouse. A Building is owned by one or more Persons. A Building can be queried for its creation date by sending it the get_creation_date() message. A Highrise building responds to the has_elevator() message, and a TownHouse does not. A person can own 0 or more Buildings. A person has an age and a name. Man and Woman are the only possible types of Persons. Men and women can be related through “marriage”. Exactly one Man and one Woman can together have any number of children, who are Persons. A Building has one or two front doors and 0 or one rear door, both of which are Doors. A Door has a width and a height. At any point in time, a Building can contain 0 or more Persons as occupants. A building must be able to reference its doors, but not the other way around. It must be possible for a Person to find out which Buildings he/she owns, but not the other way around.

(8) [5]

Draw both a collaboration diagram *and* a sequence diagram for two `Subscriber` objects `subs1` (first) and `subs2` (later) registering themselves asynchronously by means of an appropriate `subscribe` message with a `Server` object. Some time later, the server will receive a `warning("HIGH")` message from an instance of the `Monitor` class. The parameter `"HIGH"` denotes the severity level of the warning. Reception of the `warning` prompts the server to *callback* both clients in the order they registered themselves and invoke their `do_it()` methods asynchronously. While doing so, the server will make it possible for the client to know *which object* called its `do_it()` method. The `do_it()` method in *both* objects `subs1` and `subst2` uses the `help` method (with 0 arguments) in a unique object `helperSingleton`, an instance of class `Helper`.

(9) [2]

Draw a State Automaton which recognizes the regular expression `^(XY)+(C|D)D*$`

(10) [3]

List the defining features of the Statecharts formalism and draw a simple example of each.