Brock University

Final Examination, Fall 2005 — **Number of Pages: 3**
Course: COSC 2P90 — **Number of Students: 71**
Date of Examination: Dec. 21, 2005 — **Number of Hours: 2**
Time of Examination: 12:00-14:00 — **Instructors: M. Winter**

Instructions

1) **Place all answers in the examination booklet provided.**
2) **Attempt Question 8 and *any five* of Questions 1 through 7.**
3) **Unless required to write an entire class, you do not have to write `import` statements when using routines from the standard Java libraries.**
4) **You do not have to comment your code.**
5) **The paper totals 40 marks. The marks for each question are indicated on the left (e.g. [5]) and the breakdown within the question in italics (e.g. *(2)*).**
6) **No examination aids, specifically no electronic devices including calculators and electronic dictionaries, are permitted. Use or possession of unauthorized materials will automatically result in an award of zero for this examination (FHB 5.1.2.A).**
7) **A mark of 40% must be achieved on this examination in order to obtain a passing grade in the course.**

You may answer any *five* of questions 1 through 7.

[5] 1. Explain each of the following parameter passing mechanisms*(3)*: call by value, call by value-result (*not* the Algol-W version) and call by reference. Determine the value of the variable `x` after executing the following program segment if the parameter is passed using *each* of call by value-result (*not* the Algol-W version) or call by reference*(2)*:

```
var x, y : Integer;

procedure p (z : Integer);
begin
    x := 0;
    if z > y then z := 2*z else z := 2*z+1;
end;

begin
   x := 3;
   y := 1;
   p(x);
end.
```

[5] 2. Budd categorizes the most common uses of inheritance into 6 categories: specialization, specification, construction, extension, limitation and combination. Pick 3 of these categories including at least one category each of a "good" and "bad" use of inheritance. Describe the pattern of use in the category(3). Give an example of how the use is achieved in Java for two of these categories*(2)* (a description is sufficient, code need not be included).

[5] 3. Differentiate between pure polymorphism, overriding and overloading*(3)*. Describe how dynamic binding can be implemented to provide polymorphism by overriding*(2)*.

…2

[5] 4. Choose *one* of the following Design Patterns: Class Adapter, Façade and Abstract Factory. For the chosen pattern, describe the problem the pattern addresses*(1)*, give an example of the use of the pattern for a problem (you may include a class diagram)*(2)* and describe how the pattern solves the problem (you may use a class diagram of the model)*(2)*.

[5] 5. Expand the following application of a LISP function to its arguments using rewrite rules (reduction)*(4)*.

```
(DEF L (LAMBDA (X Y)
   (COND ((NULL X) NIL)
         ((EQ (CAR (CAR X)) Y) (CAR (CDR (CAR X))))
         (T (L (CDR X) Y))
   )
))

(L '((A B) (B C)) 'B)
```

In general, what does the function `Q` do*(1)*?

[5] 6. What is meant by *lazy evaluation(2)*? The following Haskell declarations compute the list `primes` of all prime numbers:

```
primes = sieve [2.. ]

sieve (p:x) = p : sieve [ n | n <- x, n `mod` p > 0 ]
```

What is the result of the application `f 3 primes` where `f` is defined as follows *(1)*:

```
f 0     (x:xs) = x
f (n+1) (x:xs) = f n xs
```

Suppose Haskell would use eager evaluation (call-by-value) and **not** lazy evaluation. What will then be the result of the application above *(2)*?

[5] 7. Explain how the cut can improve the efficiency of a prolog program*(2)*. Consider the following Prolog program:

```
person(adam).
person(jane).
person(samantha).
sellsTo(macrosoft,willmart).
sellsTo(macrosoft,adam).
sellsTo(macrosoft,jane).
sellsTo(pear,adam).
sellsTo(pear,samantha).
twoCustomersOf(X,Y,C) :- sellsTo(C,X),
                         person(X),
                         !,
                         sellsTo(C,Y),
                         person(Y).
```

…3

Show that the use of a cut in the example above is **not** appropriate (that is, it is a red cut) by giving an example where the predicate `twoCustomersOf` fails and would succeed without a cut*(3)*.

You must answer question 8.

8. A bakery wants to use a program to automate their production according to the existing orders and the availability of the ingredients needed. A regular product of the bakery has a name (`String`), a number (`int`) of existing orders and list of ingredients (each together with an amount (`int`)—the amount required to produce one product) needed to produce it. An ingredient has a name (`String`), an ingredient number (`int`) and the amount "in stock" (`int`). An ingredient may be "on order" in which case it has an expected date of shipment (`Date`) and the amount (`int`) ordered. Operations on a regular product include accessing (not updating) the name, the number of orders and the list of ingredients and in/decreasing the number of orders. Furthermore, a regular product should provide an operation `produce(n)` that first determines how many products of this kind (say `m`, from `0` to `n`) can be produced with the ingredients in stock and then decreases the number of orders for the product and the ingredients in stock according to `m`. The operation returns `m` as a result. Operations on an ingredient include accessing the name, the ingredient number, the order information attributes described above if the part is on order and accessing and updating the number in stock. Furthermore, an ingredient should provide an operation `order(n,d)` indicating that the amount `n` of the ingredient is ordered with the expected date of shipment d.

   The bakery also produces weeding cakes. A weeding cake is a product that doesn't initially have a list of ingredients. Such a cake cannot be produced until a baker has added this list of ingredients.

[8] a. Draw a UML class diagram describing the relationship between the classes described above. Show relationships, multiplicities, association names and the attributes and operations and indicate overriding of methods where appropriate.

[7] b. Give implementations in Java of the `produce` method required by regular products and wedding cakes according to the UML diagram that you have provided above.