

Date: 25-October-2007
Time: 4.30–6.30pm
Permitted Aids: None

CS246
Midterm
Fall 2007

Signature: _____

Place sticker here.

Last Name: _____

First Name: _____

ID: _____

Instructions: (Read carefully before the exam begins):

1. Before you begin, make certain that you have one Exam Booklet with pages numbered 1–13 printed double-sided.
2. The marks assigned to each question are shown at the beginning of the question; use this information to organize your time effectively.
3. Place all your answers in the spaces provided on these pages.
4. You do not need to write comments in your code unless it is specifically required by the question.
5. Questions will not be interpreted. Proctors will only confirm or deny errors in the questions. If you consider the wording of a question to be ambiguous, state your assumptions clearly and proceed to answer the question to the best of your ability. You may not trivialize the problem in your assumptions.
6. Cheating is an academic offense. Your signature on this exam indicates that you understand and agree to the University's policies regarding cheating on exams.

Question	out of	mark	marker's initials
1	12		
2	20		
3	35		
4	8		
5	15		
Total	90		

Question 1. [12 = 5 + 5 + 2] This is a C++ main program:

```
# include <iostream>
int main ()
{
    int n = 20;
    int m = 30;
    std::cout << func (m, n, m) << "," << m << std::endl;
}
```

(a) [5] Here is a definition for func:

```
int func (int x, int y, int z)
{
    int *p = new int;
    *p = x;
    if (x > y) { x = y; y = *p; }
    return x + z;
}
```

What is the output if main() is run with the above definition?

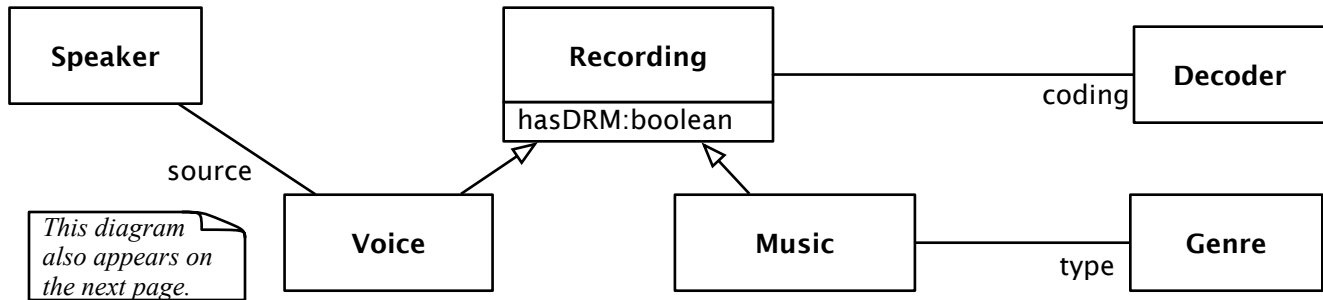
(b) [5] Here is an alternate definition for func:

```
int func (int & x, int & y, int z)
{
    int *p;
    p = & x;
    if (x > y) { x = y; y = *p; }
    return x + z;
}
```

What is the output if main() is run with the above alternate definition?

(c) [2] The definition in (a) above compiles and runs, but has a flaw: what is it?

Question 2. [20 = 5 + 5 + 10] Here is a partial domain model for a record library system. Recordings are stored on the system and must be decoded somehow to be played.

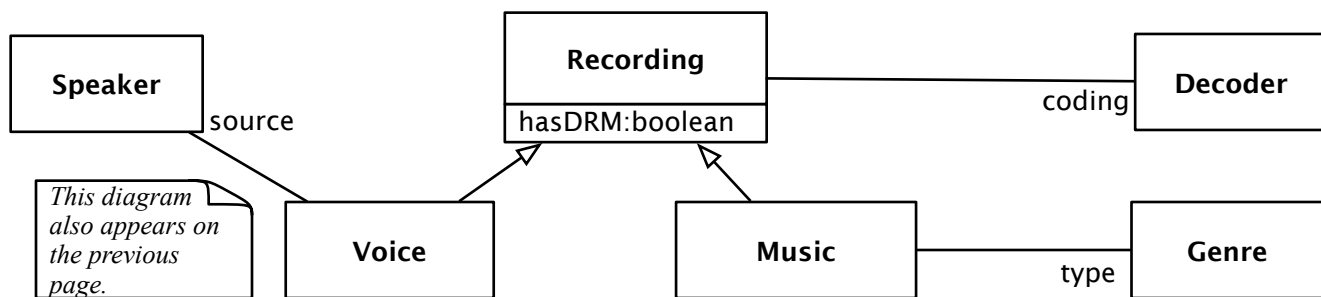


For the two statements (a) and (b) below, either explain why the statement is false, or provide an instance specification which conforms to the above model and shows why the statement is true.

(a) [5] *One voice recording can be the source of another voice recording.*

(b) [5] *A voice recording and a musical recording can have the same decoder.*

Question 2 continued.



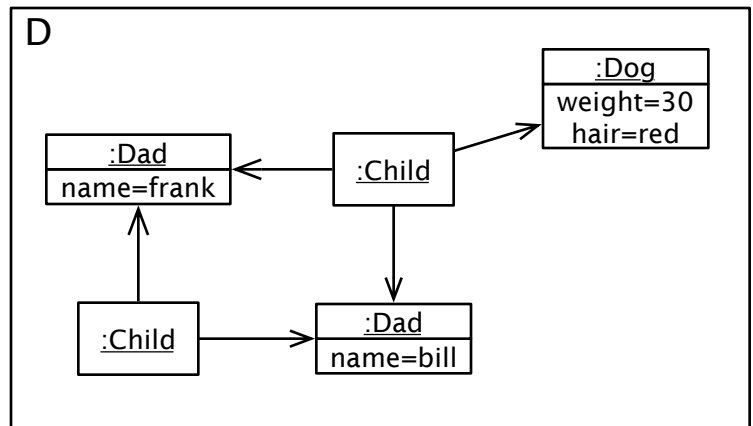
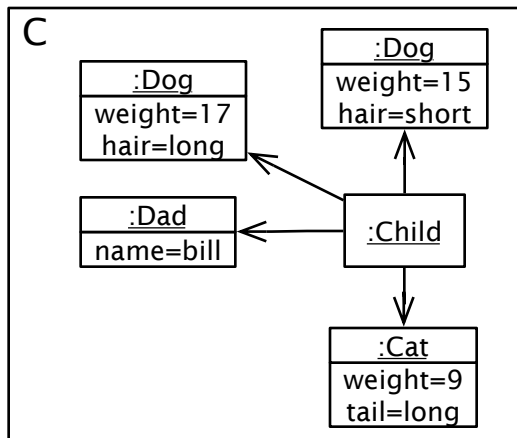
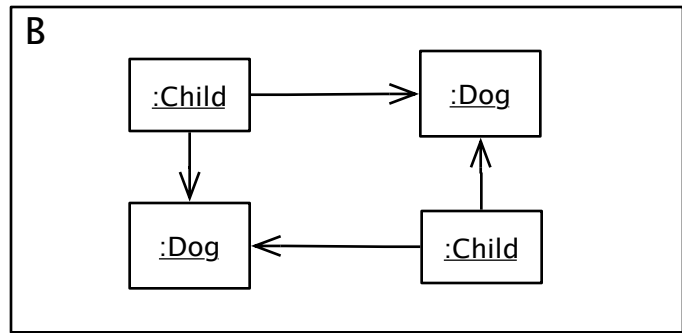
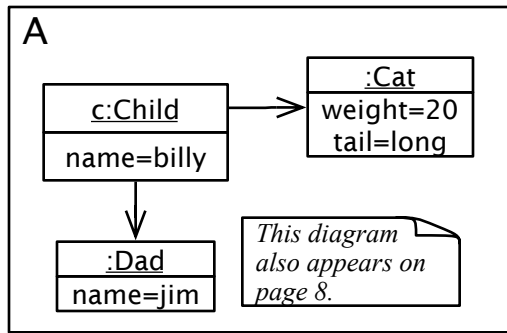
(c) [2 each] The following rules governing this domain model are to be enforced by the software:

- A recording can have at most one genre type assigned to it.*
- A voice recording has to have at least one speaker.*
- A decoder is considered a “voice decoder” if it is coding at least one voice recording.*
- A genre has a name and an optional origin, which are strings.*
- Music recordings don’t have DRM if their type name is “INDIE”.*

Redraw the model in the space below, adding any attributes, associations, multiplicities, classes, constraints, etc., that may be required so that the rules are enforced by the model.

Question 3. [35 = 10 + 15 + 10] You should read all parts of this question before attempting any of them.

Consider the following four instance models:



(a) [10] In the space below draw a class model diagram which A and B conform to, but C and D don't.

Question 3 continues...

Question 3 continued.

(b) [15] Write a complete C++ header file `Q3.h` to implement your model from (a). Each classifier is to be implemented as a C++ class. Declare class members **public**. Devise operations for navigating the associations, and declare them **public** too. *There is no need* to provide definitions (no `Q3.cc`).

Q3.h

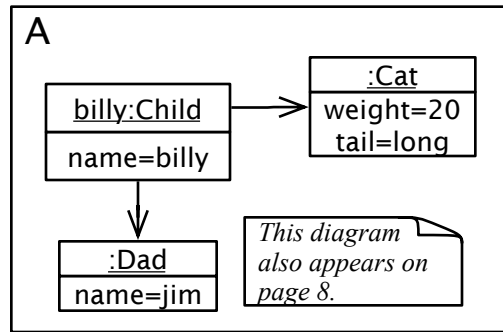
Question 3 continues...

Question 3b extra space for Q3 .h if needed.

Question 3 continues...

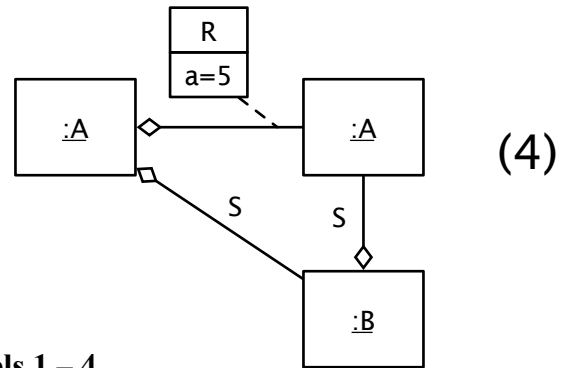
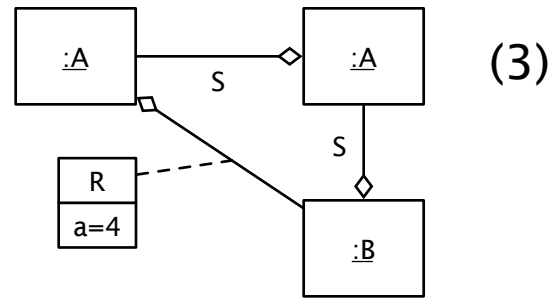
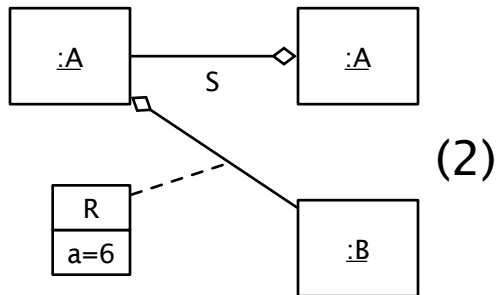
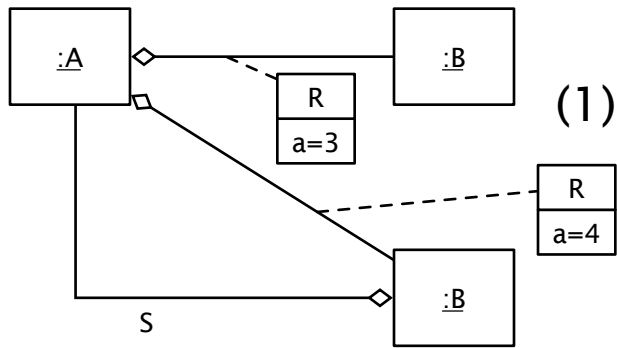
Question 3 ends on this page.

Here is diagram A again:

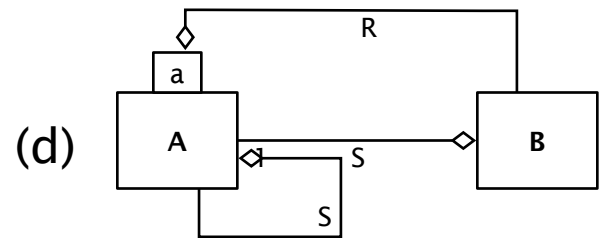
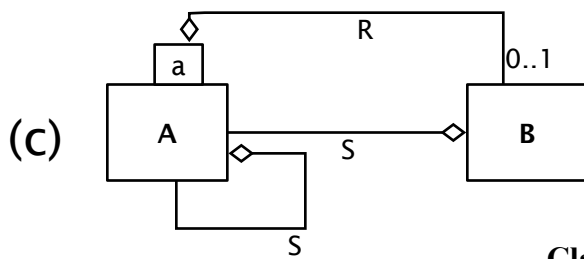
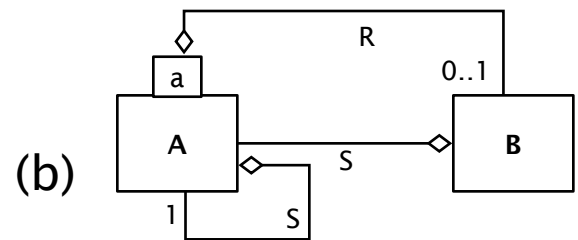
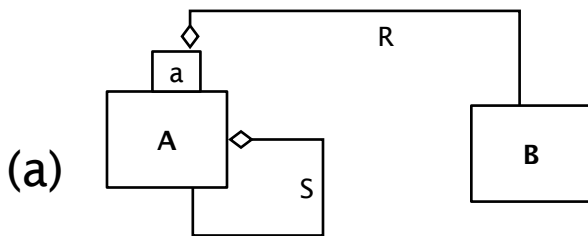


(c) [10] In the space below write a complete source file for a stand-alone function `diagramA` that uses your design from (b) to create objects as specified by the diagram A, and to return a pointer to `billy`. Use dynamic allocation for objects. Your source file should compile.

`diagramA.cc`



Instance Models 1 – 4



Class Models a – d

Question 4. [8 = 2 each] In the space below write which instance models above conform to which class models. Write NONE if an instance model conforms to none of the class models.

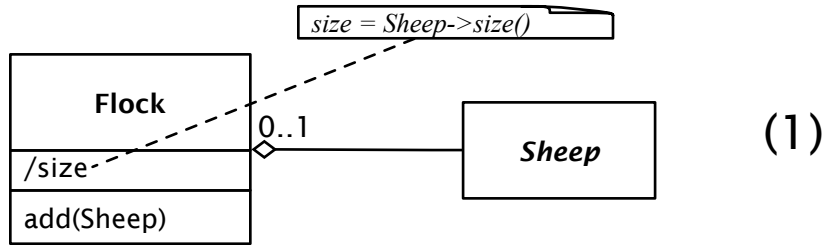
(1) conforms to :

(3) conforms to :

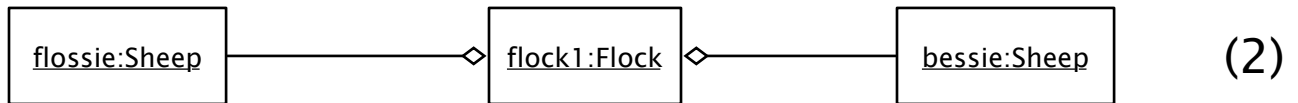
(2) conforms to :

(4) conforms to :

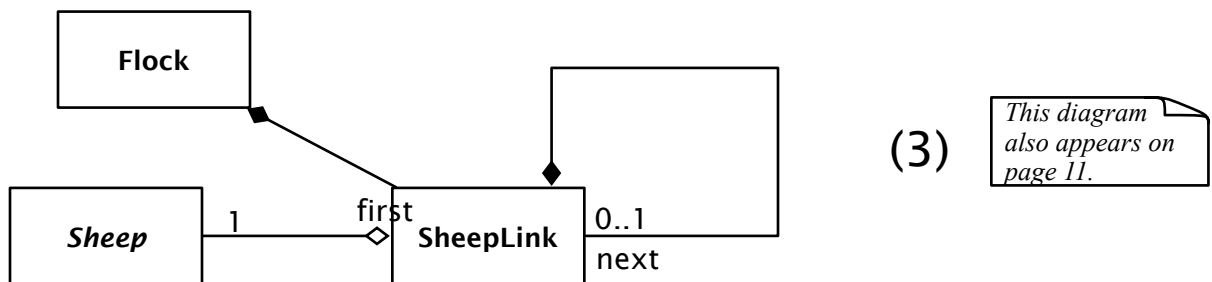
Question 5. [15 = 2 + 3 + 10] Here is a model of a Flock of Sheep. (There can be various kinds of Sheep.)



The following specifies a particular flock of two sheep conforming to the above model:



The model is to be implemented using the following linked structure:

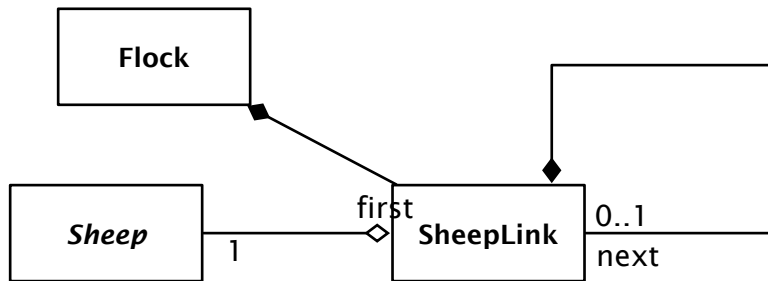


(a) [2] In the space below draw an implementation instance model (*i.e.* conforming to diagram 3) which specifies an implementation of *flock1* (*i.e.* contains the information in diagram 2).

Question 5 continues...

Question 5 continued.

Here's the implementation diagram again:



(2)

*This diagram
also appears on
page 10.*

Here are proposed C++ declarations for the implementation. Note that /size has been declared as a public method.

```
class Sheep;

class SheepLink
{
public:
    Sheep* s;
    SheepLink* next;
};

class Flock
{
    SheepLink* _first;
    int _numberOfLinks;
public:
    void add (Sheep &);
    int size ();
};
```

Here's the constructor for Flock.

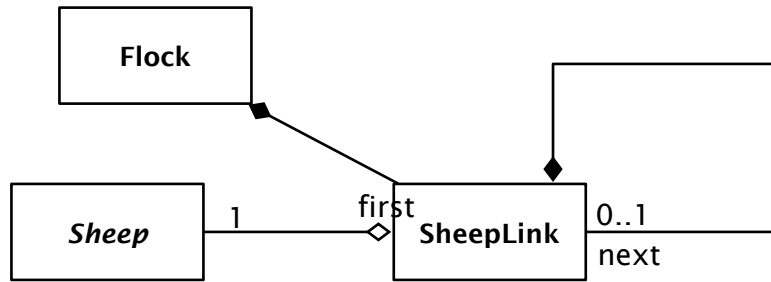
```
Flock::Flock():
    _first (0)
    _numberOfLinks (0)
{ }
```

(b) [3] In the space below write a definition for `Flock::size`.

Question 5 continues...

Question 5 ends here, finally.

Here's the implementation diagram yet again:



(3)

*This diagram
also appears on
page 10.*

(c) [10] Write a definition for `Flock::add(Sheep&)` in the space below. (Hint: Don't forget about `numberOfLinks`, and don't forget the possibility in `f.add(s)` that `s` already belongs to `f`.)

```
void Flock::add (Sheep & s)
{
```

This is extra space for any question.