University of Toronto at St. George
CSC324—Introduction to Computer Science, Spring 1999

Midterm

Duration: 50 minutes

Aids allowed: None

Make sure that your examination booklet has 7 pages (including this one). Answer in the space provided and use the reverse side of the page for rough work; **write legibly**. (If you need more space, use the reverse side of the page and indicate **clearly** which part of your work should be marked.)

Family Name: ______________________ **Tutorial room:** ______________________

First Name: ______________________ **Tutor:** ______________________

Student # : ______________________

0. ________ / 2
1. ________ / 10
2. ________ / 10
3. ________ / 10
4. ________ / 10
5. ________ / 10

Total ________ / 52

Question 0 [2 marks in total]

Write your name and student number legibly at the top of every page of this test.

Question 1 [10 marks in total]

Circle `true` for the true statments, `false` for the false ones.

1. `protected` members of class `X` can be accessed by any superclass of `X`, even though they might be in a different package.

   `true` `false`

   The answer is 'false'.

2. Packages are useful for managing polymorphism in Java.

   `true` `false`

   The answer is 'false'.

3. Java uses dynamic scoping.

   `true` `false`

   The answer is 'false'.

4. Java allows a class to inherit features from more than one abstract class.

   `true` `false`

   The answer is ambiguous, so we didn't count it.

5. To test whether an `Object` reference `x` refers to an instance of class `Broble`, it is reasonable to write `if (x instanceof Broble)`.

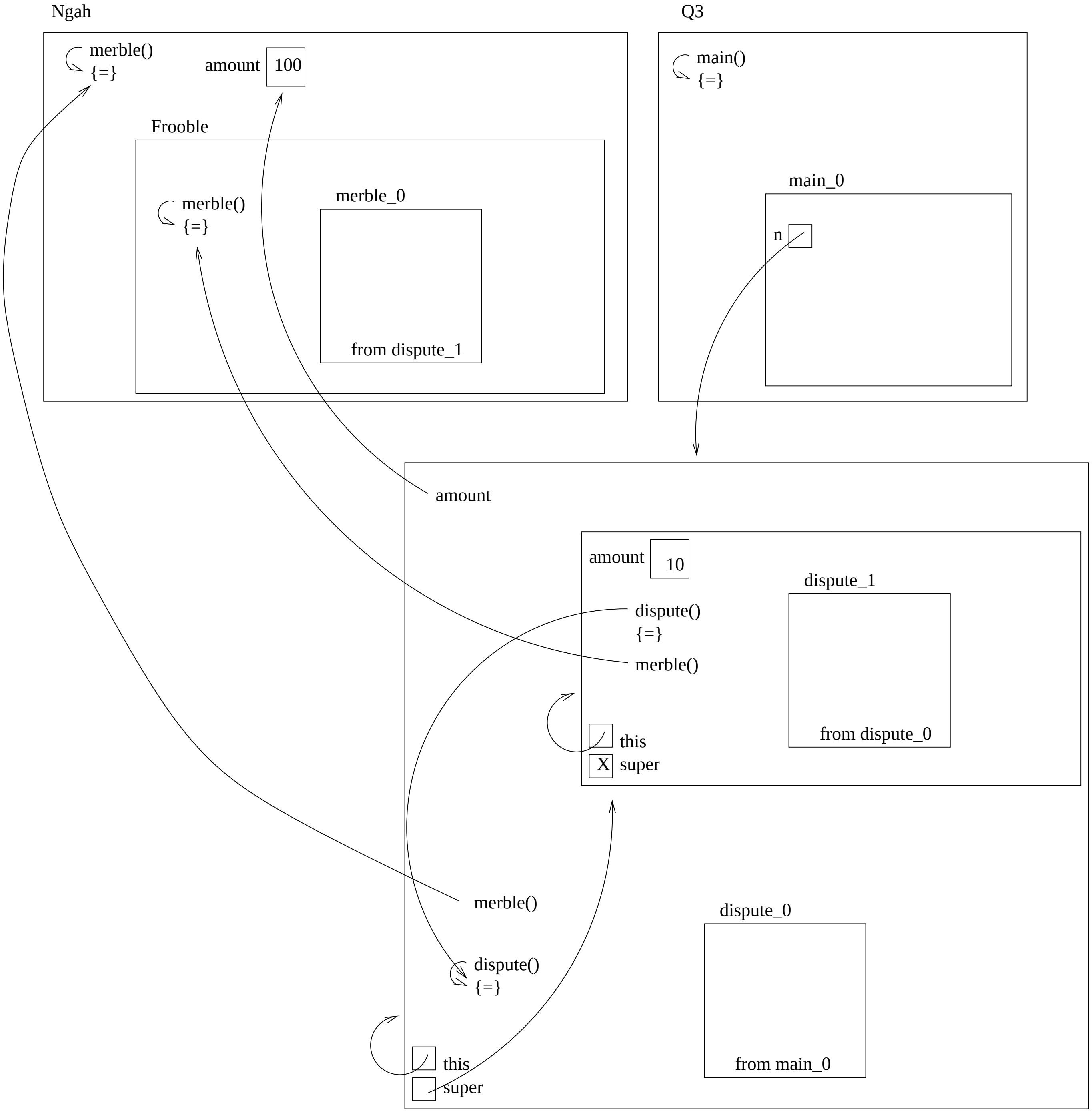
   `true` `false`

   The answer is 'true'.

Question 2 [10 marks in total]

For the following code, on the next page complete the drawing of the state of the program the first time the line marked `//***` is reached. (Note that there may be some boxes and arrows missing in the picture – draw them in. Also, don't forget to include the "from x" words to indicate the order in which method frames were called.)

```
class Frooble {             | class Ngah extends Frooble {       | class Q3 {
    int amount;             |     static int amount;             |     public static void main(
                            |                                    |             String[] pars) {
    void dispute() {        |     void dispute() {               |
        amount += 10;       |         super.dispute();           |         Ngah n = new Ngah();
        merble();           |         super.merble();            |         n.dispute();
    }                       |     }                              |         System.out.println(
                            |                                    |             n.amount );
    static void merble() {  |     static void merble( Ngah n ) { |         System.out.println(
        Ngah.amount += 100; |         n.amount += 1000;          |              ((Frooble)n).amount );
        //***               |         amount += 500;             |     }
    }                       |     }                              | }
}                           | }                                  |
```



Ngah
merble()
{=}
amount
100
Frooble
merble()
{=}
merble_0
from dispute_1
Q3
main()
{=}
main_0
n
amount
amount
10
dispute_1
dispute()
{=}
merble()
from dispute_0
this
X
super
merble()
dispute()
{=}
dispute_0
this
super
from main_0

Question 3 [10 marks in total]

Consider the following code:

```
procedure redolence (fred, wilma : integer);
begin
  fred := fred-1;
  dishIndex := dishIndex+1;
  fred := Dishes[dishIndex];
  wilma := wilma-1;
  write (fred,wilma);
  write (Dishes[1], Dishes[2], Dishes[3]);
end;
begin (* main *)
  (* Dishes and dishIndex are global vars. *)
  Dishes[1] := 10;
  Dishes[2] := 20;
  Dishes[3] := 30;
  dishIndex := 2;
  redolence (Dishes[1], Dishes[dishIndex]);
  write (Dishes[1], Dishes[2], Dishes[3]);
end;
```

Write the output for the code when the parameter-passing mechanism is call by value-result:

```
30 19
10 20 30
30 19 30
```

Write the output for the code when the parameter-passing mechanism is call by value:

```
30 19
10 20 30
10 20 30
```

Write the output for the code when the parameter-passing mechanism is call by reference:

```
30 19
30 19 30
30 19 30
```

Question 4 **[10 marks in total]**

Give an example of an ambiguous grammar. Draw a parse tree for a string generated by your grammar that demonstrates the grammar is ambiguous.

```
S ::= A | S ϵ

  S        S
  |       / \
  A      S   ϵ
         |
         A
```

Question 5 [10 marks in total]

Give the output for the following code for both static and dynamic scoping:

```
procedure Gnoble
  var i1 : integer;
  var i2 : integer;
  procedure Gnuble
    begin (* Gnuble *)
      writeln(i1, i2);
    end;
  procedure Gnable
    var i1 : integer;
    begin (* Gnable *)
      i1 := 3*i2;
      i2 := 4;
      writeln(i1, i2);
      Gnuble();
    end;
  begin (* main *)
    i1 := 10;
    i2 := 20;
    Gnuble();
    Gnable();
  end;
```

Dynamic output:

```
10 20
60 4
60 4
```

Static output:

```
10 20
60 4
10 4
```