Written Test 2

CSE 1020 3.0

Section M, Winter 2010

Family Name: ______________________________

Given Name(s): ______________________________

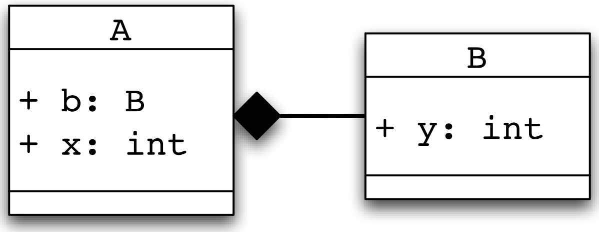
Student Number: |___ |___ |___ |___ |___ |___ |___ |___ |___ |

Guidelines and Instructions:

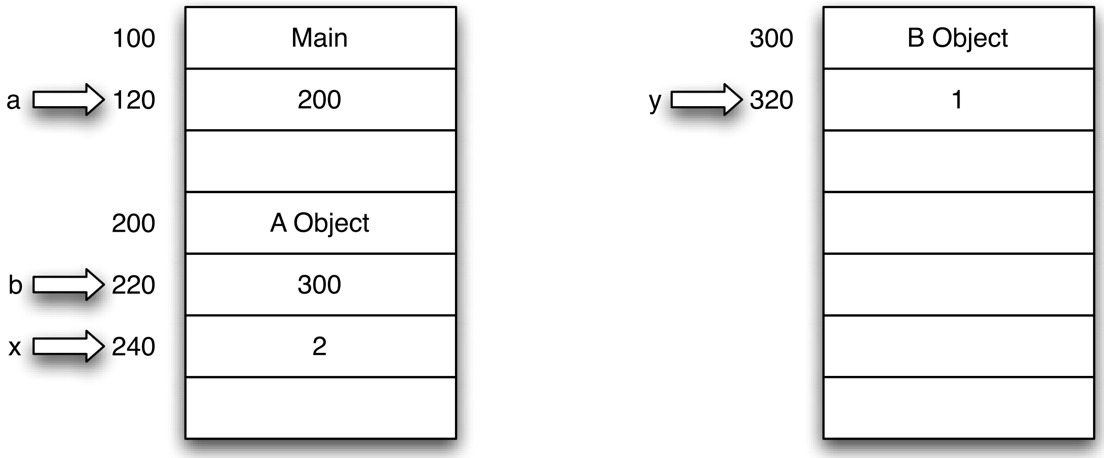
1. This is a 50-minute test. You can use the textbook, but no electronic aids such as calculators, cellphones etc.
2. Answer questions in the space provided. If you need more space, use the back of the page. Clearly indicate that your answer continues on the back of the page.
3. Write legibly. Unreadable answers will not be marked.
4. Leave your ID on the desk. A sign-up sheet will be distributed during the test. By signing it, you acknowledge that you are registered in the course and you are the owner of the associated ID.
5. Keep your eyes on your own work. At the discretion of the invigilators, students may be asked to move.

Question	Out of	Mark
Q1	32	
Q2	36	
Q3	32	
Total	100	
Letter grade		

Q1. [32 marks] Suppose that classes `A` and `B` are related as in the following diagram (only attributes are shown).



Also, consider that an app uses class `A` and has created the objects shown in the following memory diagram.



In the next page, indicate which of the following would violate the relationship between `A` and `B` shown in the first diagram. Explain clearly the reason why. Do not be concerned with implementation details, i.e. do not provide any code.

(a) Creating an alias of `a`.

(b) Creating a shallow copy of the object `a` points to.

(c) Creating a deep copy of the object `a` points to.

(d) Creating an alias of `b`.

(e) Creating a shallow copy of the object `b` points to.

(f) Creating a deep copy of the object `b` points to.

In order for any of the designated processes to violate the composition relation between classes A and B, they would need to create a situation where there is a reference to the object b points to, other than the one in the object a points to (this way the two objects will not have a shared lifetime).

The only processes that achieve this are (b) and (d). (b) will create another object of type A that points to the same object of type B, i.e. the one in address 300. (d) would create an exact copy of the contents of memory address 220 achieving a similar result but without copying the rest of the object pointed to by a.

None of the other processes creates a situation where an object of type B is pointed to by an object of type A and there is at least one more reference to it.

Q2. [36 marks] Consider the API for the two classes `Animal` and `Dog` below. For each one of the code segments in the following pages, check the box next to the expected outcome. Follow the instructions given after the selected answer. You can assume all needed classes have been imported. *No marks will be awarded without error explanation or correct output as the case may be.* Each code segment is worth 6 marks.

public class Animal

```
    This class encapsulates an animal.
Field Detail
```

public String name

```
    The name of this animal
Constructor Detail
```

public Animal(String s)

```
    The name field is assigned the value of s
    Parameters:
        s - the name of this animal
Method Detail
```

public void printName()

```
    Always prints out the string "Animal", followed by a space, followed
by the value of the name field, followed by a newline.
```

public class Dog extends Animal

```
    This class encapsulates a dog.
Constructor Detail
```

public Dog(String s)

```
    The name field is assigned the value of s
    Parameters:
        s - the name of this dog
Method Detail
```

public void printName()

```
    Always prints out the string "Dog", followed by a space, followed by
the value of the name field, followed by a newline.
    Overrides:
        printName in class Animal
```

public void bark()

```
    Always prints out the string "Arf!" followed by a newline.
```

(a)
```
Animal a1 = new Dog("Moby");
a1.printName();
```

☐ Compiles and runs (write output in space below).
☐ Compile-time error (explain source of error).
☐ Run-time error (explain source of error).

Compiles and runs. Output is:

```
Dog Moby
```

(b)
```
Animal a2 = new Dog("Moby");
a2.bark();
a2.printName();
```

☐ Compiles and runs (write output in space below).
☐ Compile-time error (explain source of error).
☐ Run-time error (explain source of error).

Compile-time error. Method `bark()` is not available for references of type `Animal`.

(c)
```
Dog d3 = new Animal("Moby");
d3.bark();
d3.printClassName();
```

☐ Compiles and runs (write output in space below).
☐ Compile-time error (explain source of error).
☐ Run-time error (explain source of error).

Compile-time error. Incompatible types in line 1.

(d)
```
Animal a4 = new Animal("Moby");
Dog d4 = a4;
d4.printName();
```

- ☐ Compiles and runs (write output in space below).
- ☐ Compile-time error (explain source of error).
- ☐ Run-time error (explain source of error).

Compile-time error. Incompatible types in line 2.

(e)
```
Animal a5 = new Animal("Moby");
Dog d5 = (Dog) a5;
d5.printName();
```

- ☐ Compiles and runs (write output in space below).
- ☐ Syntax error (explain source of error).
- ☐ Run-time error (explain source of error).

Run-time error. Class cast exception at line 2.

(f)
```
Animal a6 = new Dog("Moby");
Dog d6 = (Dog) a6;
a6.printName();
d6.printName();
```

- ☐ Compiles and runs (write output in space below).
- ☐ Syntax error (explain source of error).
- ☐ Run-time error (explain source of error).

Compiles and runs. Output is:

```
Dog Moby
Dog Moby
```

Q3. [32 marks]

(a) **[8 marks]** In the programs presented in class, we would declare and instantiate lists of strings as follows

`List<String> x = new ArrayList<String>();`

rather than

`ArrayList<String> x = new ArrayList<String>();`

Explain why the first method is preferred.

We can use the List interface without worrying about the particular implementation. The implementation can change by changing only one line in the code.

(b) **[8 marks]** One can add the string `"Java"` to a set of strings, `x`, with the statement `x.add("Java")`. This will work with either of the following declarations for `x`:

i. `Set x;`

ii. `Set<String> x;`

Which of the two declarations is preferrable and why?

2 marks - the second declaration is preferrable

4 marks - Compile-time error rather than possible run-time error if something other than String is added to the set

2 marks - No need to cast when getting elements from the set

(c) **[8 marks]** Class `Collections` contains the following method

```
static void sort(List<T> list)
```

where the following condition holds

```
T extends Comparable<? super T>
```

For each of the declarations below, indicate whether the declared collection can or cannot be sorted using this method and **why**. Class `Dog` was presented in Question 2.

i. `List<String>`
ii. `Set<String>`
iii. `ArrayList<String>`
iv. `List<Dog>`

2 marks each.

i. Yes - String implements Comparable
ii. No - A Set is not a List
iii. Yes - An ArrayList is a List and String implements Comparable
iv. No - Dog does not implement Comparable

(d) **[8 marks]** Suppose that in your program you would like to keep track of the songs that your favourite band intends to play in their next concert. Would you represent them as a set or a list, and why? If you need further information in order to make this decision, state what that would be.

Fun fact: This collection of songs is typically referred to as the band's **setlist**.

Assuming that the order songs are played is important, then a list is the right representation. However, if different assumptions are made, such as the order does not matter, but elimination of duplicates is important, then a set is the right representation.

Any answer that is justified on the basis of maintaining the element order and duplicate elimination is deemed correct.