Written Test

CSE 1020 3.0

Section A, Summer 2011

Family Name: _____

Given Name(s): _____

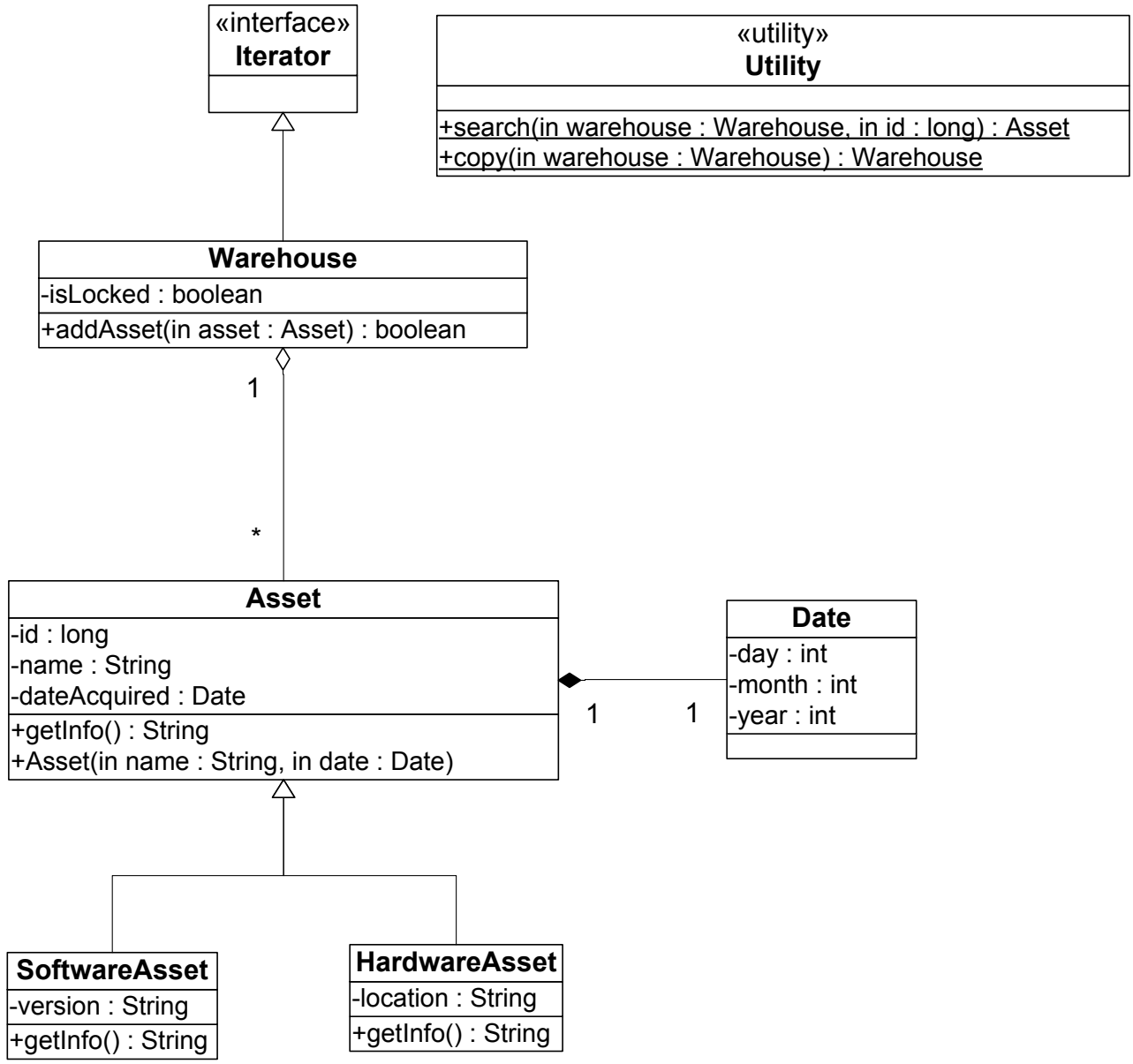
Student Number: |___ |___ |___ |___ |___ |___ |___ |___ |___ |

Guidelines and Instructions:

1. This is a 80-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	20	
Q2	20	
Q3	20	
Q4	20	
Total	80	
Letter grade		

All questions in this exam refer to the following set of classes. Study them carefully before answering any questions.



Some of the attribute and methods in these classes are self-explanatory. The following pages provide extra information on the remaining attributes as well as the `getInfo`, `search` and `copy` methods.

The `Warehouse` class stores information about company assets. The `isLocked` attribute

indicates whether the warehouse is locked, which means that it is impossible to add assets.

The company has two types of specialized assets: `SoftwareAsset` and `HardwareAsset`

Every asset has a `name`, an automatically generated unique `id` number, and a purchase date (`dateAcquired`).

Software assets have the attribute `version` which contains information about the software version, such as `"Microsoft Word 2008"`.

Hardware assets have the attribute `location` which contains information about the location of the asset, such as `"Server Room"`.

The `getInfo` method in class `Asset` returns a string containing detailed information about the asset as in the sample below:
Type : Asset
ID : 12
Name : Windows
Date Acquired : 14/07/2011

The `getInfo` method in class `SoftwareAsset` returns a string containing detailed information about the software asset as in the sample below:
Type : Software
ID : 12
Name : Windows
Version : Windows 7
Date Acquired : 14/07/2011

The `getInfo` method in class `HardwareAsset` returns a string containing detailed information about the hardware asset as in the sample below
Type : Hardware
ID : 15
Name : Switch
Location : Server Room
Date Acquired : 14/07/2011

The `search` method in class `Utility` finds and returns an asset that matches a given `id`. If no asset matches the given `id`, `null` is returned.

The `copy` method in class `Utility` create a copy of the `Warehouse` object.

Instructions pertaining to all questions in this exam

- Minor syntax errors and outputting formatting problems in your answers **will not** affect your mark. Concentrate on concepts rather than syntax.
- Classes `Warehouse`, `Asset` ,`SoftwareAsset`, `HardwareAsset`, and `Date` have constructors that initialize all attributes, as well as accessors and mutators for all attributes

Q1. [20 marks] Assume that `w` is a reference to a `Warehouse` object that is already populated with assets.

(a) [10 marks] Write a fragment that prints detailed information about all assets in `w`. The datailed information must be as in Pg. 3, i.e. it differs depending on whether we have a generic asset, a software asset, or a hardware asset. No marks will be given for unnecessarily convoluted solutions.

```
for(Asset a:w)
{
  output.println(a.getInfo());
}
```

(b) [10 marks] Write a fragment that calculates how many hardware assets are in `w`.

```
int count = 0;
for(Asset a:w)
{
  if (a instanceof HardwareAsset)
    count++;
}
```

Q2. **[20 marks]** Write a fragment of code which determines what level of copy is implemented in the `copy` method of the `Utility` class. Your fragment must output one the following strings as appropriate: `Aliasing`, `Shallow`, `Deep`.

```
Warehouse w = new Warehouse();
Date date = new Date("12/12/12");
Asset asset = new SoftwareAsset("test","version",date);
w.addAsset(asset);
Warehouse w2 = Utility.copy(w);
if (w2 == w)
   output.println("Aliasing");
else {
    Asset a  = Utility.search(w, asset.getID());
    Asset b  = Utility.search(w2, asset.getID());
    if (a == b)
        output.println("Shallow");
    else
        output.println("Deep");
}
```

Q3. [20 marks] Write a fragment that verifies the relations between classes `Asset`, `Warehouse`, and `Date` in the UML diagram. If all is well, the fragment should output nothing. Otherwise, it should output one or both of the following messages as appropriate:

```
There is no composition between Date and Asset
```

```
There is no aggregation between Warehouse and Asset
```

```
Date date = new Date("12/12/12");
Asset asset = new SoftwareAsset("test","version",date);
date.setDay(14);
if (asset.getDateAcquired().equals(date))
       output.println("There is no composition between Date and Asset");
Warehouse w = new Warehouse();
w.addAsset(asset);
asset.setDateAcquired(new Date("12/12/12"));
Asset b = Utility.search(w,asset.getID());
if (!b.equals(asset))
    output.println("There is no aggregation between Warehouse and Asset");
```

Q4. **[20 marks]** You are asked to test an application that reads two integers, x and y, and outputs a string. The key part of the code is shown below:

```
final int MIN = 5;
final int MAX = 10;
final int LIMIT = 5;
if (x < MIN)
{
        output.println("Left");
} else if (x < MAX && y < LIMIT)
{
   output.println("Middle");
} else if (x < MAX)
{
        output .println("Top");
} else
{
        output.println("Right");
}
```

Propose test cases that will accomplish execution-path coverage.

Execution-path coverage include test cases which ensure that every statement and every path in code will be executed. (0, 0) ; (6,1); (6;7); (12,0);