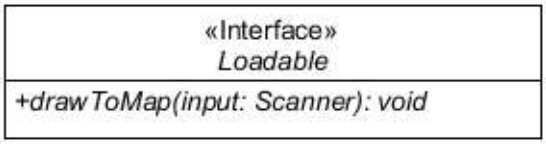
CS 249: Assignment 8

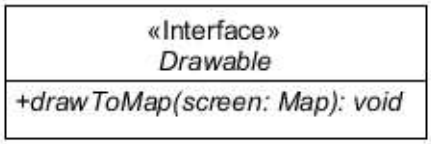
Abstract Classes and Interfaces

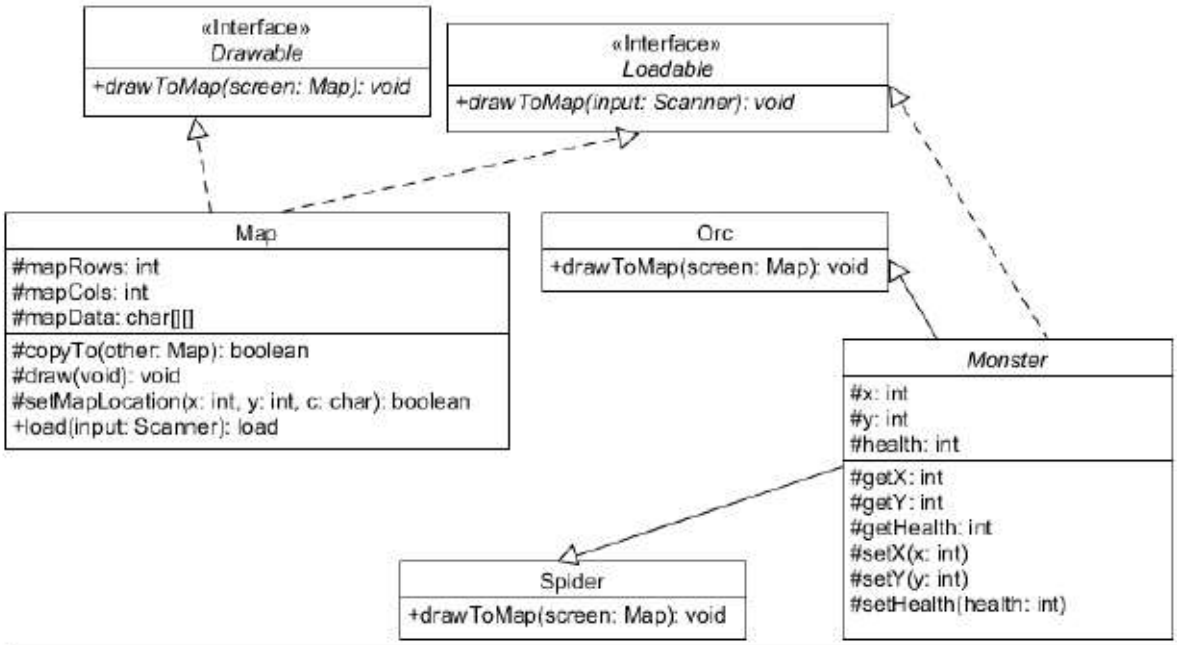
Theory Questions (16%)

1. (2%) Draw the UML diagrams for the following:



- **Interface: Loadable**
  - *Abstract methods:*
    - void load(Scanner input) throws Exception
- **Interface: Drawable**
  - *Abstract methods:*
    - void drawToMap(Map screen)





2. (2%) Draw the UML diagrams for the following.

SHOW THE RELATIONSHIPS BETWEEN THESE CLASSES AND THE INTERFACES Loadable AND Drawable. If a method has a concrete implementation, list it in the class diagram (e.g., draw() for Orc and Spider).

- **Abstract Class: Monster**
  - implements Drawable and Loadable
  - Data fields:
    - int x (default: 0)
    - int y (default: 0)
    - int health (default: 100)
  - Methods:
    - PROTECTED No-arg constructor (does nothing)
    - PROTECTED constructor that takes x, y, and health
    - Getter/setter methods for all data fields
    - Concrete implementation of Loadable methods
    - (NO concrete implementation of Drawable)
- **Class: Orc**
  - extends Monster
  - Data fields: None
  - Methods:
    - No-arg constructor (does nothing)
    - Constructor that takes x, y, and health (calls super constructor from Monster)
    - Concrete implementation of Drawable
- **Class: Spider**
  - extends Monster
  - Data fields: None
  - Methods:
    - No-arg constructor (does nothing)
    - Constructor that takes x, y, and health (calls super constructor from Monster)
    - Concrete implementation of Drawable

3. (2%) Given an **abstract class** BakedGood, is the following code legal? If not, why not?

```
BakedGood b = new BakedGood();
```

No, abstract classes cannot have implementations.

4. (2%) Given an **abstract class** BakedGood and a class Muffin that extends BakedGood, is the following code legal? If not, why not?

```
BakedGood b = new Muffin();
```

b is implementation of Muffin class

5. (2%) Given an **interface** Edible and a class Muffin that implements Edible, is the following code legal? If not, why not?

```
Edible e = new Muffin();
```

Interfaces should be implemented in Class declaration.

6. (2%) A class can implement more than one Java interface.

   (a) True
   (b) False

(a)True

7. (2%) A Java interface can have private methods.

   (a) True
   (b) False

(b) False

8. (2%) All data fields defined in a Java interface are public, static, and final.

   (a) True
   (b) False

(a) True