Practise Quiz #2:

Note: The corresponding questions on Quiz #2 (to be written the week of Oct 31) will test the same material as these questions, although the format will be different.

Quiz #2 will cover: Inheritance, Shadowing and Overriding, UML, Generics, Exceptions, and CRC cards.

1.(a) The following code refers to class `Student` which extends class `Person`:

```
Student suying = new Student;
System.out.println(suying.toString());
int age = yufei.x;
```

Assume that both the `Person` and the `Student` class contain **different** variables that are both integers called `x` and both contain **different** methods called `toString()`.

Explain what the following table means, using the `Person` and `Student` classes as an example:

	Static	**Instance**
Variable	shadow	shadow
Method	shadow	override

yes. up-casting always allowed

Can we cast `suying` as a `Person`?

If we include the line "`Person alfonso = new Person;`", can we cast `alfonso` as a `Student`? Why or why not?

throws ClassCastException, because it is a Person

1.(b) On the course website, under "Lectures and Labs", there is a file called `Grade.java`. Write a concrete class called `GradeOutOfFive` that extends `Grade`. Which method(s) need to be defined in `GradeOutOfFive`? Are we allowed to write an alternate version of `toLetter()` in our subclass, even though it is static in `Grade`?

just gpa() method required for subclass

A static method in abstract class cannot be abstract.
we can make a static method in subclass. However it will hide the superclass method.

2.(a) This question refers to the same `Grade.java` file as in 1(b), as well as the two child classes that can be found in `LetterGrade.java` and `NumericGrade.java` respectively. Draw a uml diagram representing these three classes and their relationships. Include a dotted arrow pointing away from the diagram for any class(es) that implement `serializable`, but do not include a box for `serializable`. We will be learning about it after the quiz.

```
@Override
  public String concat(T[] array){
    Character[] a = array;
    String s = "";
    for(Character chr: a){
        s = s.concat(chr.toString());
    }
    return s;

  }
```

2.(b) Write a generic method for `Grade.java` that takes in an array of values of type `T` as input and returns a single string that includes all values in the array with spaces between them. Then implement this method twice so that `T` represents objects of type `Character` in class `LetterGrade` and objects of type `Integer` in class `NumericGrade`.

3. In Java, `Throwable` is a class that has direct subclasses `Error` and `Exception`. Search the internet for documentation on class `Exception` before doing this question.

   (a) Choose five of the listed subclasses of `Exception` (e.g., `InvalidTargetObjectTypeException`) and write a code fragment for each one that catches the exception in an appropriate context.

   (b) Write your own subclass of `Exception`. When would you want it to be thrown? Write a code fragment that throws it.

checked exception is useful in that it resolves exceptional conditions and recover from it without terminating the program

   (c) Under what circumstances is throwing a subclass of `Exception` more useful than throwing a subclass of `RuntimeException`? What does "unchecked" mean in this context?

unchecked: internal exceptions that application cannotanticipate or recover from

4. You are developing software to facilitate the operation of a lending library. The library lends out items that can be classified as tools or media. There must be a way to report tools that no longer function properly with a description of the problem and a rating of "useable" or "not useable". There should be a way to submit reviews of media as well as a numerical rating out of five. Administrative users of the system should be the only people with access to the tool reports. But everyone who logs in with a password should have access to the media reviews. The system should also allow people to place holds on currently checked out tools and media, but not without a password. However, no password should be required to view the current inventory of tools and media and whether or not each item is currently in the library or not.

   Follow the steps from the slides and handout on CRC to create CRC cards to break up the above scenario into interfaces and/or classes with appropriate methods and variables.