Tutorial 4

These questions refer to the lectures presented in week 3 of Object Oriented Programming with Java. They will be covered during the tutorial during week 5.

Relationships

1. Give examples of Association, Aggregation and Composition involving Car, Wheels, ParkingLot, and Driver.
2. What is directionality for associations?
3. What is cardinality for associations?
4. Give examples of uni-directional and bi-directional associations involving Car, Wheels, ParkingLot, and Driver.
5. Give examples of one to one, one to many, and many to many associations involving Car, Wheels, ParkingLot, and Driver.
6. How do you create a uni-directional association?
7. How do you create a bi-directional association?
8. How do you implement a one to one relationship?
9. How to you implement a one to many relationship?
10. How do you implement a many to many relationship?
11. How do Java arrays differ from C arrays?
12. Draw the UML representation of the association, aggregation and composition defined in question 1. Include cardinality.

```
package oop;

public class CarPark {

    private Car[] cars;

    public CarPark() {
        cars = new Car[0];
    }

    public Car findCar(int lotNum) {
        return cars[lotNum];
    }

}
```

ArrayList

13. What advantage to an ArrayList give over an Array?

14. Refactor the class CarPark defined above to use an ArrayList.
15. Do you change the unit test when refactoring? Why or why not?
16. Why is it complicated to change the interface to a class?
17. What is the first step in changing an interface to a class?

Generics

18. What is a Generic Class?
19. How do you specify a parameter to a generic class?
20. What are two advantages of using generics?
21. How do you define a generic class?
22. How many parameter types can you add to a generic class?
23. How do you specify a generic method?
24. Can generic method exist in non-generic classes?
25. What is a static method? How is it called?