Assignment 5

Student Name:	Grader Name:
Student UIN:	Grader UIN:

Reading Assignment: How to Program Java

- Chapter 17 – Files, Streams and Object Serialization
- Chapter 18 – Recursion
- Chapter 19 – Searching, Sorting and Big O
- Chapter 20 – Generic Collections
- Chapter 21 – Generic Data Structures
- Chapter 22 – Custom Generic Data Structures

True or False:

1. Typical I/O operations are very slow compared to the speed of accessing data in memory.
2. Java imposes no predetermined structure on a file; one must structure files to meet the needs of an application.
3. An `Iterator` is often used to cycle through a collection.
4. Once a `Vector` is initialized, its size cannot be changed.
5. Values of primitive types can be stored directly in a collection.

Short Questions:

1. What is the difference between an **absolute path** and a **relative path**?
2. What does it indicate when an argument of 0 is passed to the `System.exit` method?
3. Why are sequential-access file not updated in place usually?
4. What is a **serialized object**?
5. Explain the difference between iteration and recursion.

Programming Challenge: Suppose that a generic Android application is getting information from sensor modules that are connected through bluetooth. The code needed to interface with networked sensors and microcontrollers is being written by a remote team. Your team is responsible to use the class produced by the remote team to populate a local database that contains sensor readings over time.

Define a Java `interface` that specifies a suitable data format for the information pulled from the sensors, along with a set of relevant methods. Describe desirable data types using, possibly, class `Collections`. In particular, although sensors may have different structures, data could be acquired in an efficient, polymorphic way. Write an abstract class that can be extended and implemented in different sensing scenarios.

Explain the rational for your design decisions in less than 500 words. The result from command

```
prompt> wc -w $filename
```

should be less than 500. That is, be concise.