[answers online]

Name:_____

Student ID:_____

Total points: 110

1. What do you think of the **Java** programming language overall? Compare it to another language you know. **[4]**

2. What is **object-oriented** programming? How does it differ from procedural programming? **[4]**

3. Describe and contrast the four options we have in Java for **access modifiers** on attributes/methods. **[4]**

4. What is the effect of the **`final`** keyword on
   - (a) instance variables? **[2]**
   - (b) methods? **[2]**
   - (c) classes? **[2]**

5. What is an **interface** in Java? What is an **abstract class**? What's the difference? Describe an example of each (you may come up with your own). **[6]**

6. Name and briefly describe the features of at least 4 **Swing widgets** (i.e., subclasses of JComponent). If you can't remember the exact name of the class, describe the widget in as much detail as you can. **[8]**

7. What is object **serialization** in Java? Describe a situation where serialization might be useful. **[4]**

8. What is a **component framework**? Why are they cool? Describe one such framework in detail (it need not even be computer-related). Describe a couple example components and what you could do with them. **[5]**

9. What is an Android **Activity**? Is it a complete application? Describe an example. **[4]**

10. Why do Android applications not have a **main()** method? **[3]**

11. What are Android **string resources**? Why are they cool? [4]

12. Design a **class hierarchy** of your own choosing, with an abstract superclass and at least two concrete subclasses.

    (a) Design attributes and methods appropriate for each class and draw a UML **class diagram**. [5]

    (b) Design a method that takes advantage of the **polymorphism** in this class hierarchy, and explain how it does so. The polymorphic method need not live inside any of the classes. [4]

13. For each of the three categories of **design patterns** in the "Gang of Four" book, name at least three example patterns (for a total of nine patterns). For each pattern you choose, describe it in words and give an example or analogy illustrating it.

    (a) **Creational**: [5]

(b) **Structural**: [5]

(c) **Behavioural**: [5]

14. Write a complete Java program that reads in a file "input.txt", removes all **digits**, and outputs the result in the file "output.txt". Include full **doc-comments** with pre-/post-conditions. [8]

15. A **`BankTransaction`** consists of an account number, date, dollar amount, and a flag that indicates whether it was a withdrawal or deposit.

    (a) Write a complete Java **class** defining a BankTransaction, including attributes and two constructors (one with a full set of parameters, and one with no parameters at all). Be sure to do error-checking of all input parameters. To keep things simple, no set/get functions are required.
    The `java.util.Date` class stores a date/time object; the default constructor returns the current date/time. [6]

    (b) Add a **copy constructor** to the implementation. Check for null references and other errors. [4]

16. Design a console-based networked **chatroom** in Java. Users may be able to join and leave the chatroom, and any line of text entered by any user is broadcast to all other users. What **classes** will you need? What **methods** might those classes have? Describe how you would handle having **multiple users** connected at the same time. Describe in detail what should happen when a new user tries to connect to the chatroom. [8]

17. We wish to design a system that tracks the location of **public buses** in real-time and updates displays at bus stops informing riders of how long they'll need to wait for the next bus. Draw a UML **use-case** diagram for this system. [8]