CIS 121 - Exam 2 Solution - Tuesday, April 6, 2004

Name: ______________________________ J Number: ______________________________

1. Fill in the Blank (30 pts)

1. LIFO means **Last In First Out** and describes the behavior of a stack.

2. The keyword **throw** causes an exception to occur.

3. Java supports a **termination** model of exception handling.

4. A **tag interface** contains no methods and acts as an indication to the compiler and interpreter that a class definition has a certain capability.

5. We create a logical group of JRadioButtons by creating an instance of a **ButtonGroup** and adding the JRadioButtons to it.

6. Java's **catch-or-declare rule** states that when a method contains code that might cause a checked exception to occur we must either inform the compiler of which exceptions can occur or deal with them ourselves.

7. FIFO means **First In First Out** and describes the behavior of a simple queue.

8. When reading from a BufferedReader with the read method, we know we are done reading when the method returns **-1**.

9. The method setPreferredSize requires a **Dimension** object as parameter.

10. We call the data stored in files **persistent data** since it can still be present when a program exits.

11. When reading from a BufferedReader with the readLine() method, we know we are done reading when the method returns **null**.

12. A **throws clause** normally contains a list of checked exceptions a method might throw.

13. The operation of placing a new element on the top of a stack is called a **push**.
14. SQL mean [Structured Query Language].

15. When using the method readObject from the ObjectInputStream class we do not only have to deal with the possibility of an IOException but also with the possibility of a [ClassNotFoundException].

16. When we break a String into pieces based on a delimiter, we call those pieces [tokens].

17. The unchecked exceptions are [RuntimeException] and [Error] and their subclasses.

18. The superclass of all exceptional behaviors in Java is [Throwable].

19. The method [size()] determines how many elements are in a Vector.

20. The method executeQuery produces a [ResultSet] object.

21. We can obtain a reference to a [URLConnection] object by calling the method openConnection on a URL object.

22. In the data hierarchy, [bytes] are collections of bits, and records are collections of [fields].

23. The method [getCodeBase()] gives us a reference to the URL where an applet’s .class file is found.

24. The process of converting an object into bytes which can be written to a file is called [serialization].

25. True/False False When we create a FileWriter object using the constructor with one String parameter, if the file we are trying to create already exists, its contents will not be overwritten.

26. We can use the method [toLowerCase()] to change all of the characters of a String to lower case.

27. Regardless of what object reference is required, the reference [null] can always be used.

28. The operation of removing an element from a queue is called [dequeuing].
II. Matching (10 pts) Please choose the most appropriate choice for each term from the candidates on the right.

30. next() N B. A superclass of benign exceptions that can occur during execution which can usually be handled by the programmer
31. (int)c where c is a char D C. A superclass of serious exceptions that can occur during execution
32. Serializable J D. Gives the ASCII value of a character
33. “Hello”.substring(1) T E. Returns a reference to a String array with 6 elements
34. Loads an AudioClip in a JFrame A F. A method in the ResultSet class that returns a value associated with a row in the ResultSet
35. Error C G. A variable that we can never place in a class definition
36. throws P H. Returns a reference to a String containing 1
37. serialVersionUID UID O I. Makes an external connection to a file if we intend to read bytes from it
38. FileInputStream I J. An interface which is implemented by classes so that we can write objects created from it to a file

K. Causes an exception to happen
L. Makes an external connection to a file if we intend to read characters from it
M. Converts a character to a random integer
N. A method in the ResultSet class that returns a boolean
O. A variable created for a class definition that doesn’t have one if it implements an interface that allows objects created from the class to be written to a file
P. Precedes a list of checked exceptions that a method might cause to happen
Q. An interface containing 5 methods that must be implemented before we can create objects from the class

R. Returns “Hello”
S. getAudioClip
T. Returns “ello”
III. Short Answer (30 pts)

39. (4 pts) Suppose we have an Image called image. Show one line of code which will cause image to point to a scaled version of itself where the width is 1/2 of the original width, and the height is 1/2 of the original height.

```java
image = image.getScaledInstance(image.getWidth(this)/2, image.getHeight(this)/2, Image.SCALE_DEFAULT);
```

40. (4 pts) Suppose the String literal “TTTTTTttttTtttTtttSTTtTtttTTttTTttTT” is referred to by the reference string. Show one line of code which will convert this String into the String literal “test”.

```java
string = string.toLowerCase().replaceAll("t+", "t");
```

41. (2 pts) Does changing a String reference to refer to another String violate the immutability of Strings? Explain.

No, it doesn’t. A String reference is not a String, it is a reference to a String.

42. (2 pts) Is a finally block ever required when we use a try block? Explain.

Yes. If a try block doesn’t have any catch blocks, it must have a finally block.

43. (3 pts) What are three methods we can use to place an element into a Vector?

- `addElement`
- `insertElementAt`
- `setElementAt`

44. (4 pts) What is the data hierarchy?

- `bits`
- `bytes`
- `characters`
- `fields`
- `records`
45. (4 pts) Suppose we have a BufferedReader object called input, and an array of chars called chars. Suppose that the chars array already has 10 characters in the first part of the array, and we don’t want to disturb them. Show one line of code that will read in characters from input and attempt to fill in the remaining elements chars array.

```java
read(chars,10,chars.length-10);
```

46. (2 pts) What are two ways of determining if there are more tokens to be read from a StringTokenizer?

```java
hasMoreTokens()
compare countTokens() to 0
```

47. (2 pts) Suppose that the String reference s refers to the String “Hello”. What does s refer to after the line containing only `s.toUpperCase().replaceAll("H","h").toLowerCase().substring(1);` is executed? Explain.

“Hello”. Since Strings are immutable, and we don’t change s to point to another String, s is still pointing to “Hello”

48. (1 pt) Is there a syntax error with the following line. If so, what is the error?

```java
throw new String("\n");
```

**Yes. You can only throw an object which is Throwable**

49. (1 pt) What will the following print? Explain.

```java
System.out.println("1;2;3;4;5;6".split(";")["1;2;3;4;5;6".split(";").length]);
```

2. When we split the first String, it produces an array with 6 elements. When we split the second String, there is no instance of the delimiter, so there is only one token. So what will print is the element from the first array at position 1

50. (1 pt) What three streams are created whenever we execute a Java program?

```java
System.in
System.out
System.err
```
IV. Problem Solving and Coding (30 pts)

51. (5 pts) What is printed by the following code? Make sure you read the code carefully.

```java
public class Question51 {
    public static void main(String[] args) {
        String s = "This is a test";
        int index = s.indexOf(‘‘);
        while (index > -1) {
            System.out.println(s.substring(0, index) + " " + index);
            index = s.indexOf(‘‘, index + 1);
        }
    }
}
```

This is a test

52. (5 pts) Suppose we want to write a method to remove the repeated elements from a String array and return the result. Fill in the missing pieces in the following code. Assume the java.util package has been imported.

```java
public String[] removeRedundancies(String[] array) {
    Vector v = new Vector();
    for (int counter=0; counter<array.length; counter++)
        if (!v.contains(array[counter]))
            v.addElement(array[counter]);
    return ((String[])v.toArray(new String[0]));
}
```

53. (5 pts) Recall what a try-catch-finally block does, what will be printed by the following code. Is there any syntax error in this code? Note that print is used instead of println.

```java
public class Question53 {
    public static void main(String[] args) {
        String s = "1;2;3;4;5;6;Hello;7;8;9;10";
        String[] temp = s.split(";");
        for (int counter=0; counter<temp.length; counter++) {
            try {
                int number = Integer.parseInt(temp[counter]);
                System.out.println(number);
            } catch (NumberFormatException npe) {
                System.out.print(1);
            } finally {
                System.out.print(".");
            }
        }
    }
}
```

1-2-3-4-5-6-1-7-8-9-10-
54. (8 pts) Write a small class definition called MyClass containing at least one instance variable called number. Include an instance method in this class definition which will write the object that calls the method to a file whose name is specified as the parameter to the method. Also include a class method which will return an object of type MyClass which will be read from a file whose name is specified as the parameter to the method. Be sure to include all necessary code. You may not use any throws clauses with the methods.

```java
import java.io.*;

public class MyClass implements Serializable {
    private int number;

    public void store(String fileName) {
        try {
            FileOutputStream file = new FileOutputStream(fileName);
            ObjectOutputStream output = new ObjectOutputStream(file);
            output.writeObject(this);
            output.close();
        } catch (IOException ie) {
        }
    }

    public static MyClass read(String fileName) {
        try {
            FileInputStream file = new FileInputStream(fileName);
            ObjectInputStream input = new ObjectInputStream(file);
            MyClass myClass = (MyClass)input.readObject();
            input.close();
            return(myClass);
        } catch (ClassNotFoundException cnfe) {
            return(null);
        } catch (IOException ie) {
            return(null);
        }
    }
}
```

55. (7 pts) Suppose we have an applet containing a group of JCheckBoxes stored in an array called boxes. Assume all of these JCheckBoxes have added the applet as their ItemListener. Suppose we also have two AudioClips in the class definition called selected and deselected. Show the complete method inside the applet's definition that will be called when one of the JCheckBoxes is selected or deselected and will play the appropriate AudioClip based on whether or not a JCheckBox was selected or deselected.

```java
public void itemStateChanged(ItemEvent e) {
    if (e.getStateChange() == ItemEvent.SELECTED)
        selected.play();
    else
        deselected.play();
}
```