PRINT YOUR NAME: ___________________________ UFID [5:8]: ________

I have not looked at anyone else’s paper, and I have not obtained unauthorized help in completing this exam. Also, I have adhered to and upheld all standards of honesty as stated in the University Honesty Policy and in the course syllabus.

YOUR SIGNATURE: ___________________________ DATE: _________________

YOU ARE ALLOWED TO HAVE ONLY THE FOLLOWING ON YOUR DESK OR WORKTABLE:

1. A *HANDWRITTEN* CRIB SHEET 8-1/2 x 11 inches (ONE SIDE ONLY)

2. A BASIC CALCULATOR – NO GRAPHING, NO PROGRAMMABILITY

3. YOUR PEN OR PENCIL AND AN ERASER

ALL OTHER OF YOUR BELONGINGS BROUGHT INTO THE CLASSROOM MUST BE ON THE FLOOR, UNLESS YOU REQUIRE EYEGLASSES OR AN OPTICAL MAGNIFYING GLASS.

PLEASE TURN ALL CELL PHONES, iPHONES, iPADS, AND OTHER ELECTRONIC DEVICES (EXCEPT CALCULATORS), OFF AND PUT THEM AWAY – RINGTONES AND CELL PHONE CONVERSATIONS NOT PERMITTED DURING EXAM.
This exam has four regular questions and one extra-credit question. Complete the questions that are easiest for you first, then complete what you can of the difficult questions. There is no penalty for guessing. However, on questions involving calculation, you must show your work. If you do not show your work, you risk getting only partial credit for any answer.

Q1. (30 pts) Define the following terms using 1-3 sentences or a formula (6 pts each):

(a) **Inheritance:** inheritance (in Java) is when a class extends another class, and consequently includes all the visible methods and variables from the other class as part of its own interface.

(b) **Polymorphism:** the ability to create a variable, object, function, operator, or object that has more than one form; also defined as the ability for values of different data types to be handled using a uniform interface.

(c) **Superclass:** the parent of a subclass. Provides functionality, in the form of methods and variables, that are inherited by a subclass.

(d) **Subclass:** a class which extends a superclass. Inherits functionality, in the form of methods and variables, from the superclass.

(e) **Abstract Class:** a class which declares an abstract method – a method that lacks an implementation (body). Abstract classes cannot be implemented directly. They must be first extended by subclasses, which provide implementations of the abstract methods.
Q2. (45 pts) Given the concepts of INHERITANCE and POLYMORPHISM, answer each of the following two questions completely (20 pts for 2.1, 25 pts for 2.2):

2.1) Suppose you have a class *FunClass* with public methods *show*, *tell*, and *smile* and private methods *get* and *put*. If *MoreFunClass* is a subclass of *FunClass*, then write the Java statement that creates this subclass by inheritance:

Write Java Code: 
```java
class MoreFunClass extends FunClass {
}
```

List the methods that *MoreFunClass* has (explain why each one is in *MoreFunClass*):

- *show*, *tell*, *smile*: these methods are inherited and accessible from *FunClass* because *MoreFunClass* extends *FunClass*, and because these methods are public in *FunClass*. *MoreFunClass* also includes all the methods of java.lang.Object, because all classes extend Object.

2.2) What are two ways to create polymorphism in a programming language (we discussed this in class several times). Write your answer here:

(1) Make a method (or function) for each class (or datatype) that operates on that datatype but has the same name across all datatypes; (2) make a function that is inherited from a superclass that is truly polymorphic, operating on all instances and subclasses of that superclass.

Pick one polymorphic operator and write a java code fragment that demonstrates its function for ints, floats, doubles, and strings. Also indicate what function the operator is performing in the table below.

<table>
<thead>
<tr>
<th>Code Fragment</th>
<th>Role</th>
</tr>
</thead>
</table>
| int | int x = 5, y = 6;  
|     | int z = x + y;     | The + operator is performing 32-bit integer arithmetic. |
| float | float x = 5.0f, y = 6.0f;  
|      | float z = x + y;     | The + operator is performing single precision floating point arithmetic. |
| double | double x = 5.0, y = 6.0;  
|       | double z = x + y;     | The + operator is performing double precision floating point arithmetic. |
| String | String x = "5", y = "6";  
|        | String z = x + y;      | The + operator is performing string concatenation. |
Q3. (30 pts) Answer the following questions (write clearly 1-2 sentences, 6 pts each):

a. What kind of class cannot be instantiated as an object? Why?

Abstract classes cannot be instantiated because they may have one or more methods that lack an implementation.

b. What is the highest-level superclass in the Java language? What does this mean?

java.lang.Object. This means that all instances of all classes inherit the methods of Object, and can be stored in an Object reference.

c. What is a constructor and how is it used in Java?

A constructor is a method that is used to create a new instance of a class. The constructor of FooClass is invoked by "new FooClass(…args…)"

d. Explain the primary mechanism for code re-use in object-oriented programming?

Inheritance provides code re-use because it allows all specializations of a class to share a single implementation of methods which are defined in the superclass.

e. What is an interface in Java, and how is it used?

An interface is a list of methods that a class which implements the interface must implement. Classes may implement multiple interfaces. This is useful because it allows an object to be stored in a variable whose type is any of the interfaces it implements, which is typically used for message passing and event handling (callbacks).

Q4. (35 pts) Given a vector of integers \( v = (1, 2, 3, 4) \), write Java code for inserting these numbers into (a) an Array, and (b) an ArrayList (DECLARE ALL VAR’s):

<table>
<thead>
<tr>
<th>Array Code</th>
<th>------------------</th>
<th>ArrayList Code</th>
<th>-----------------</th>
</tr>
</thead>
<tbody>
<tr>
<td>int[] arr = new int[]{1,2,3,4}; OR</td>
<td>int[] arr = new int[4];</td>
<td>ArrayList&lt;Integer&gt; al = new ArrayList&lt;Integer&gt;();</td>
<td></td>
</tr>
<tr>
<td>arr[0] = 1;</td>
<td>arr[1] = 2;</td>
<td>al.add(1);</td>
<td></td>
</tr>
<tr>
<td>arr[2] = 3;</td>
<td>arr[3] = 4;</td>
<td>al.add(2);</td>
<td></td>
</tr>
<tr>
<td>al.add(3);</td>
<td>al.add(4);</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Extra Credit Problem:

EC. (25 pts) Given (name, number) pairs $A = \{(Bob, 4), (Amber, 5), (George, 3), (Will, 7)\}$, write a Java class with method for (1) inserting these pairs into a pair of ArrayList structures called $NameAL$ and $NumberAL$, then (2) displaying the pairs to the screen as “My name is: ” <name> “and I am in Year “ <number> “ at University.” You can choose what to name your class and method, but your solution should form a complete class that can be compiled using $javac$.

*Hint:* In Assignment 4, we stored tuples of data in an ArrayList using a special wrapper class. This is the opposite approach – storing each member of the tuple in a separate ArrayList.

PLEASE COMMENT YOUR CODE SO WE KNOW WHAT IT IS SUPPOSED TO DO (and so we can give full credit for a correct result).

```java
import java.util.ArrayList;
public class ExtraCredit{
    public static void myMethod(){
        //create the ArrayLists
        ArrayList<String> NameAL = new ArrayList<String>();
        ArrayList<Integer> NumberAL = new ArrayList<Integer>();
        //add elements to the list
        NameAL.add("Bob");
        NumberAL.add(4);
        NameAL.add("Amber");
        NumberAL.add(5);
        NameAL.add("George");
        NumberAL.add(3);
        NameAL.add("Will");
        NumberAL.add(7);
        //print out the elements
        for(int i=0; i < NameAL.size(); i++){
            System.out.println("My name is: "+NameAL.get(i)+
            "and I am in Year "+
            NumberAL.get(i)+ " at University.");
        }
    }
}
```