True or False: 20 points (2 points each)

1.) True or False: Accessing random elements is more efficient when using a list than when using an array.

2.) True or False: \((5 > 15 \land \text{list.isEmpty()}) \lor (9 > 8)\)

3.) True or False: Interfaces cannot be instantiated in Java.

4.) True or False: A variable declared with the `public` access modifier can only be referenced within the class which declared it.

5.) True or False: A class that has abstract methods must be declared as abstract.

6.) True or False: The body of a `do / while` loop executes 1 or more times.

7.) True or False: A class can implement multiple interfaces.

8.) True or False: A `break` statement can be used to cause a loop to terminate.

9.) True or False: A variable of type `long` can store larger integers than a variable of type `int`.

10.) True or False: If no cast is used, a `double` can be assigned to an `int`, but an `int` cannot be assigned to a `double`.

Fill in the Blank: 50 Points (5 points each)

11.) ______________ is a way of solving a problem by breaking it into progressively simpler problems until a base case is reached.

12.) ______________ are statements which repetitively execute another statement (or a block of statements).

13.) A programmer maintains ______________ by preventing the internal representation of a class from being exposed.

14.) The principle of ______________ allows programmers to use a class by its published interface without having to understand how it is internally implemented.

15.) Variables and methods that are declared with the ______________ modifier are shared by all instances of a class, and can be referenced or invoked without an instance.
16.) We call the ______________ to instantiate a new instance of a class.

17.) Polymorphism, or the ability of a variable to store many different types of related objects, is made possible by ______________ and ______________.

18.) ______________ define the state of a class's instances.

19.) ______________ define the behavior of a class's instances.

20.) When choosing access modifiers, we can choose from public, protected, default (not specified), and ______________.

Multiple Choice: 50 Points (5 points each)

21.) What output will the following code produce?

    int a = 5;
    int b = 2;
    double c = a/b;
    System.out.println("The answer is " + c);

   a.) 0.0  
b.) 2.5  
c.) 1.0  
d.) 2.0

22.) The following piece of code will cause a compile-time error. Why doesn't Java let us perform this assignment? Select all that apply.

    double a = 5.5;
    int b = a;

   a.) The assignment of a to b is widening.  
b.) The assignment of a to b requires a cast.  
c.) The assignment would cause data loss.  
d.) The assignment will round the value of b to the integer value 6

23.) The precedence of the operators in the following statement is:

    int theScore = 5 * (scores[0] + 1);

   a.) * [] + =  
b.) [] + * =  
c.) = [] * +  
d.) * + [] =
24.) The problem in the following piece of code is:

```java
int[] scores = new int[10];
for (int i=0; i<100; i++)
  System.out.println("Score " + i + ": " + scores[i]);
```

a.) The array `scores` is not initialized before it is referenced.
b.) The code will exceed the bounds of the array.
c.) `scores` will be null when printed.
d.) The code will not compile.

25.) Select all types that capable of storing are floating point values:

a.) boolean  
b.) double 
c.) float 
d.) char  
e.) int

26.) Select all that apply. With the enhanced for loop Java allows programmers to:

```java
for (Object o : objects) {
  // ...
}
```

a.) Iterate over all elements of an array.
b.) Recursively access all objects in a data structure 
c.) Use any Object implementing the `java.util.Iterator` interface to iterate. 
d.) Invoke each method of an object in the order they are declared.

26.) What is the logical error in the following code fragment?

```java
Object o;
System.out.println(o.toString());
```

a.) The `Object` class does not define the `toString` method.  
b.) The Object reference has not been initialized.  
c.) Object must be referenced with its fully qualified name `java.util.Object`

27.) Select all that apply. The `import` statement:

a.) Must be the first non-comment line in a Java file. 
b.) Allows us to use library classes and classes from different packages in our program. 
c.) Can be used with the wildcard (*) to import all classes in a package. 
d.) Traditionally comes after the class declaration, but before the package declaration.
28.) Select all that apply. Methods known as *mutators*:

a.) Must always be private.
b.) Can help maintain encapsulation by ensuring a value to be assigned to instance variables is valid.
c.) Should be avoided in program design.
d.) Allow external entities to invalidate an object’s internal state.

29.) Select all that apply. A *stack* is a data structure that:

a.) Forces data to be accessed in a First In First Out manner.
b.) Forces data to be accessed in a Last In First Out manner.
c.) Can be implemented with a list.

30.) Select all that apply. In Java, objects:

a.) Are used to encapsulate state.
b.) Can be created with special static methods known as *instantiatiors*.
c.) Are instances of classes.
d.) Inherit implicitly from primitives.

Short Answer: *60 Points (15 points each)*

31.) In a few sentences, explain the benefits of using a data structure such as a list instead of an array.

32.) I am writing a Library program, with the classes Library, Book, and Person. If I find that a Book is already checked out when requested, I want to be able to store people on the wait list for the Book in a fair way (first come, first serve). What data structure should I use to store this information and why?

33.) In a few sentences, explain the purpose of access modifiers and how they help encapsulation.

34.) Explain the distinction between the `==` operator and the `equals` method, and when each should be used.
Principles in Practice: 40 Points (20 points each)

This is meant to evaluate your understanding of programmatic control structures and object relationships needed to solve problems in Java. You will not be penalized for minor syntax errors.

35.) Using recursion or iteration, complete the following method without using the * operator.

```java
public int multiply (int one, int two) {
}
```

36.) Using the following class definitions, finish implementing the insert method for SortedIntegerList, which sorts ints from smallest to largest. Use the back of this page if you need to.

```java
public class IntegerNode {
    IntegerNode next;
    int number;

    public IntegerNode (int number) {
        this.number = number;
    }
}

public class SortedIntegerList {
    // (closing bracket omitted)
    protected IntegerNode head;

    public SortedIntegerList () {
        this.head = null;
    }

    public void insert (int number) {  // (closing bracket omitted)
```