The completed exams will be available in Student Pick-up in the Math/Science division office after the beginning of Winter quarter. The division office is the south end of the second floor in the Instructional Building (IB). Ask any staff person working in the office for your exam.
1. What is the typical source for the value(s) for the parameter(s) of the application method?

The following 2 questions are related.

2. Write a simple Java class that will represents a sphere with the following two properties: radius and volume. You may omit the JavaDoc comments. The radius of an instance of a sphere is fixed; it cannot be changed once the instance is created. The radius is given as the parameter to the constructor. Here is the formula for the volume of a sphere:

\[ \text{volume} = \frac{2}{3} \pi r^3 \]

3. Recalling the categories of properties we saw in the Properties lab, what kind of property is volume. Briefly explain.
4. Based on the JavaDoc comment, implement the following method, using the methods listed on page 9. Extra-credit for not using the `get` method of List.

```java
/**
 * Given a List of floats, the posMaxMin method returns the minimum and maximum positive values from the list in a two-element array, with the minimum positive value as the first element and the maximum positive value as the second element. When there are no positive elements in the List, the return array contains two zeroes.
 * @param vals The List to be searched.
 * @return The array with the minimum and maximum positive values of the given List.
 */
```
The following 2 questions are related.

5. List the five (5) primary flow-of-control (branching and looping) statements of Java. Use the keyword that starts a statement as the name of the statement.
   - Branching statements
   - Looping statements

6. Give simple examples of the syntax for three of the statements listed in the previous question. (For extra credit, indicate optional parts or alternate forms of the statements.)

7. The `break` statement can be used: (mark all that apply)
   a) to end processing within an exception handler.
   b) to interrupt normal program flow to start debugging.
   c) to prevent “fall through” in a `switch` statement.
   d) to terminate the execution of a `while` loop.
   e) to provide time for the user to get a cup of coffee.

8. The `continue` statement can be used: (mark all that apply)
   a) to end a single iteration within a `for` loop.
   b) to resume normal program flow after a `break` statement.
   c) to provide a `no-op` statement for a label.
   d) to exit a `then` or `else` part in an `if` statement.
   e) to jump to the loop test in a `do` loop.

9. The `return` statement can be used: (mark all that apply)
   a) to continue program flow back at the point an error occurred.
   b) to specify a return value for a method.
   c) to force a line-break in textual output.
   d) to exit a method.
   e) to indicate that the user has come back from getting coffee.
The following 2 questions are related.

10. Write a method named `equalSpace` that takes three parameters: an `int` (n) and two `doubles` (lo, hi). The method will return an array of n `double` values. The array will contain equally spaced values between the endpoints, lo and hi. Include the JavaDoc comment.
   For example, `equalSpace(5, 2.0, 8.0)` returns the array `{2.0, 3.5, 5.0, 6.5, 8.0}`.

11. Assume there is a `printArray` method that takes one parameter, an array of `doubles`, and print it to `System.out`, as follows:
    
    `{2.0 : 3.5 : 5.0 : 6.5 : 8.0}`
    
    Call the `equalSpace` and `printArray` methods to print out the following:
    
    `{5.0 : 3.0 : 1.0}`  (Extra credit if you do this in one line of code.)
The following three (3) questions are related.

12. Briefly describe the effect of the **final** modifier on methods.

13. Briefly describe the effect of the **static** modifier on methods.

14. Can a method be both **final** and **static**? Explain

15. In the Tic-Tac-Toe assignment and the ButtonButton demo and lab, you used ActionListener to handle button clicks. Complete the three steps that you need to complete to handle button clicks:
   
   a. Create a class that
   
   b. Write a method that
   
   c. Call the _____________________________ method (from the standard library)

16. java.util.Map is a subinterface of java.util.Collection. True or False Explain the reason. (An answer without an explanation will receive zero points.)

17. Briefly describe the two ways the Java keyword **super** is used in Java source code. (For partial credit, describe the significance of the keyword.)
18. The relationship between Java classes can be described using the terms *Is-A* and *Has-A*. Briefly define these two relationships. Partial credit for giving examples of the two relationships.

   Note: An answer of “yes” or “no” without an explanation will earn zero points.

20. List the possible kinds of contents (members) of a Java interface. Be as explicit and complete as possible.

21. Briefly describe the differences between exceptions and syntax errors. (Note use of plural.)
blank page for the sake of pagination
Here is a list of potentially useful methods from the List interface in the standard Java library:

```java
public boolean add(E e);
public void add(int index, E e);
public void clear();
public boolean contains(Object o);
public E get(int index);
public int indexOf(Object o);
public boolean isEmpty();
public Iterator<E> iterator();
public int lastIndexOf(Object o);
public E remove(int index);
public boolean remove(Object o);
public int size();
public List<E> subList(int fromIndex, int toIndex);
```