1. There is a Java primitive type that has exactly two values. Name the type and give the values.

2. The class **Point2D** has a single constructor that takes two *int* parameters, the x-, and y-coordinates in a Cartesian coordinate system. The parameter values range between -1024 and 1023, inclusive. The **Point2D** class has a parameterless methods named *getR* and *getTheta* that return *double* values, the radius (r) and theta (θ) coordinates for the point in polar coordinates. Create an instance of **Point2D** and call the method *getTheta*. Store the value returned in the variable *theta* declared below. (For extra credit, create the point somewhere along the line \( y = 3x+2 \).)

   ```java
   double theta;
   ```

For the next five (5) questions, write “None” if there is no such keyword.

3. What is the Java keyword associated with the methods not returning a value?

4. What is the Java keyword associated with reference variables not accessing an instance?

5. What is the Java keyword associated with the current instance?

6. What is the Java keyword associated with the base class?

7. What is the Java keyword associated with instantiation?
Examine the code for the class that appears on page 9. Assume the code is correct, that is that it compiles and “runs”. Based on this code, answer questions 8 – 19. If there is nothing to list, write the word “None”.

8. List any local variables that are declared in this class. Give the declaration (type and name). Include the line numbers where they are declared.

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>line number</th>
</tr>
</thead>
</table>

9. List any parameters that are declared in this class. Give the declaration (type and name). Include the line numbers where they are declared.

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>line number</th>
</tr>
</thead>
</table>

10. List any fields that are declared in this class. Give the declaration (type and name). Include the line numbers where they are declared.

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
<th>line number</th>
</tr>
</thead>
</table>

11. List any methods that are declared in this class. Give the method header. Include the line numbers where they are declared.

<table>
<thead>
<tr>
<th>header</th>
<th>line number</th>
</tr>
</thead>
</table>
12. List any constructors that are declared in this class. Give the constructor header. Include the line numbers where they are declared.

<table>
<thead>
<tr>
<th>header</th>
<th>line number</th>
</tr>
</thead>
</table>

13. List the reserved words used in this class. Include all the line numbers where they are used.

<table>
<thead>
<tr>
<th>keyword</th>
<th>line number</th>
<th>keyword</th>
<th>line number</th>
</tr>
</thead>
</table>

14. List the methods called by this class. Give the name of the method, the name of the class the method is in, and the line number where the call occurs.

<table>
<thead>
<tr>
<th>method name</th>
<th>class</th>
<th>line number</th>
</tr>
</thead>
</table>

15. List any fields of other classes that are accessed in this class. Give the name of the field, the name of the class the field is in, and the line number where the access occurs.

<table>
<thead>
<tr>
<th>field name</th>
<th>class</th>
<th>line number</th>
</tr>
</thead>
</table>
16. List any arguments that appear in this class. Give the name of the method or class if in a constructor, the argument, and the line number where it appears.

<table>
<thead>
<tr>
<th>method / class name</th>
<th>argument</th>
<th>line number</th>
</tr>
</thead>
</table>

17. List the classes that are used in this class. (The other classes that must exist for this code to compile.)

18. List any overloaded constructors or methods. Provide complete headers for the overloads.

19. Is there any indication of inheritance in this example? If so, which classes are related and how. (Be specific: indicate which is superclass and which is subclass.)

20. What is a tag? Where can they appear?
21. The following source code has several errors in it. Circle each error. For each error, describe the source of the error and how to fix it. Notes: Comments are part of the code. There is a penalty for circling things that are not errors. If there is one error and you make four changes to fix it, it still only counts as one error.

```java
import java.awt.Point;

/**
 * This is the Midterm class. It has several errors.
 * @author Dan
 * @version Autumn 2014
 */
public class Midterm {

    // The data for the Midterm class.
    private Point location;

    /**
     * This makes a new Midterm instance.
     * @param pt The value for the instance variable.
     */
    public Midterm(Point location) {
        location = location;
    }

    /**
     * Retrieves the value of the instance variable.
     * @param location The current location
     */
    public point getLocation() {
        return location;
    }

    /**
     * This clones the Midterm instance.
     * @return The clone of the Midterm instance.
     */
    public Midterm Midterm() {
        Midterm clone = new Midterm();
    }
}
```
22. Give the output for the following application. Print the output to the right of the fragments. The area to the left is for scratch work.

```java
public class ReadCode {

    int i, j, k;

    public static void main(String[] args) {
        ReadCode here = new ReadCode();
        here.a();
        here.b();
        here.c();
    }

    public void a() {
        System.out.println("Method a:");
        i = 0;
        j = 0;
        do {
            j += ++i;
            System.out.println(i + " - " + j);
        } while (j < 6);
        System.out.println(i + " - " + j);
    }

    public void b() {
        System.out.println("Method b:");
        i = j = k = 7;
        for(int i = 2; i < 11; i += 3) {
            j -= 2;
            k += j;
            System.out.print(i + " - " + j);
            System.out.println(" - " + k);
        }
        System.out.print(i + " - " + j);
        System.out.println(" - " + k);
    }

    public void c() {
        System.out.println("Method c:");
        double j = 1;
        i = 1;
        while(i < 10) {
            j += i / 3;
            System.out.println(i + " - " + j);
            i += 2;
        }
        System.out.println(i + " - " + j);
    }
}
```
23. There are a limited number of Java primitive types. List all of them and indicate which types can automatically convert to another type using an arrow (type_a \rightarrow type_b). Use the minimum number of arrows. For extra-credit, give the name of this process of automatic conversion of primitive Java types.

24. Briefly define the terms Is-A and Has-A. That is, how are these terms used in programming? Give an example of each relationship.

25. Update the **MyClass** class so it supports a property named **hue** of type **Color** that we have used in NscWindow assignments. Add get and set methods for the property. Use a field you declare to store the value for the property. You do not have to include comments.

```java
public class MyClass {

```
26. Write a Java method that will return the volume of a cone. It takes two parameters: radius of the base ($r$) and the height of the cone ($h$). No need to include the rest of the class, just the single “naked” method. Include the appropriate JavaDoc comment.

\[ volume = \frac{1}{3} \pi \cdot r^2 h \]

27. When you compile Java ________________, you get ________________ that runs in the ________________. The file extension of the input to the compiler is ________________; the file extension of the output of the compiler is ________________.

28. Give the header for the method which must appear in every Java program.

29. Compare and contrast method overriding and method overloading? Explain how they are similar and how they differ.
Use the following Java code to answer the questions on pages 2 through the middle of page 4.

```java
public class abc {
    public mug wag;
    private key asdf = new key();
    public abc(key jar) {
        if (jar.open < 17)
            wag = key.color(jar.rgb);
        else
            wag = jar;
    }
    public key abc(mug bap) {
        glob yada = bap.length(this);
        return yada;
    }
    public abc() {
        wag = new mug(asdf.rgb());
    }
}
```