Section 3 - Overriding Methods, Polymorphism, and Static Classes Quiz Answers

1.

Method overloading is useful because you can have multiple methods with the same signature, but different parameters. This lets you create methods with so-called *optional parameters*.

For example:

```
public void operateOnPerson(Person person, boolean isOld)
{
	// Code to operate on a person
}

public void operateOnPerson(Person person)
{
	operateOnPerson(person, false);
}
```

Method overloading lets us specify default behavior – in this case, if the programmer calls the second operateOnPerson method, then they're assuming that the person isn't old.

2.

The static keyword is useful for accessing global information. If we have a variable that should be shared throughout our program, then it doesn't make sense to have to construct objects to access that variable. The same logic applies to static methods.

3.

Polymorphism and inheritance work much better than does the instanceof operator in terms of code maintainability and readability. Use instanceof when it's impossible or very difficult to use polymorphism, such as when you can't modify the base source code.

4.

You can't subclass a class that is declared final.

5.

```java
public static double average(Double... list)
{
      double average = 0;
      for (Double d : list)
      {
            average += d;
      }

      return average / list.length;
}
```

Notice how this is entirely equivalent to considering the list parameter as an array of doubles.