1) Which of the following statements are correct with respect to the diagram shown?

```java
public abstract class A {
    ...
}
public class B extends A {
    ...
}
public class C extends A {
    ...
}
```

A. B and C are subclasses that A extends.
B. A is a superclass that B and C extend.
C. B and C are superclasses that extend A.
D. A is a subclass that extends B and C.

2) Which of the following does not apply to an abstract class:

A. Has a constructor
B. May have some concrete methods
C. May have some abstract methods
D. Constructor not used for instantiation of abstract class
E. Required to contain abstract methods

3) Select the correct definition of the usage of a Java abstract class.

A. An abstract class provides shared code and data for a set of classes that share attributes and behaviors.
B. An abstract class is similar to an interface in that it specifies functionality, but has no actual code or data.
C. An abstract class differs from an interface in that it must implement every method that it contains.
D. An abstract class can be instantiated, but code for its abstract methods might be missing.
5) Which of the following does not apply to a concrete class:

A. Has a constructor
B. Has only concrete methods
C. Has some abstract methods
D. Constructor used for instantiation
E. May have instance variables

6) Which of the following does apply to an interface (pre-1.8):

A. Has a constructor
B. Has some concrete methods
C. Has some abstract methods and some concrete methods
D. Has only abstract methods
E. A class may only implement 1 interface.

7) Which of the following does not apply to a class that implements Comparable:

A. Must implement compareTo
B. Must implement equals
C. Allows for a natural ordering of objects
D. Can be stored in a variable of type Comparable
E. Can be stored in an array of type Comparable

8) What does the following print?

```java
public class CompareTo {

    public static void main(String[] args) {
        java.util.Date date1 = new java.util.Date(2013, 1, 1);
        java.util.Date date2 = new java.util.Date(2012, 1, 1);
        System.out.println(date1.compareTo(date2));
    }
}
```