This quiz is to be taken by yourself with closed books, closed notes, no calculators.

Given the following definitions:

```java
public interface Speakable {
    public abstract String speak();
}
```

```java
public class Puppy implements Speakable {
    private static final String PUPPY_SPEAK = "Woof";

    public Puppy() {
        // ctor initialization here
    }

    public String speak() {
        return PUPPY_SPEAK;
    }

    public String wag() {
        return "wag tail";
    }
}
```

```java
public class Kitty implements Speakable {
    private static final String KITTY_SPEAK = "Meow";

    public Kitty() {
        // ctor initialization here
    }

    public String speak() {
        return KITTY_SPEAK;
    }

    public String sleep( int time ) {
        return time + " minute cat nap";
    }
}
```

And the following variable definitions:

```java
private Puppy puppy;
private Kitty kitty;
private Speakable speakable;
```

What gets printed with the following statements (each statement is executed in the order it appears). If there is a compile time error, write "Error".

```java
puppy = new Puppy();
kitty = new Kitty();
speakable = kitty;
System.out.println( speakable.sleep( 10 ) );
System.out.println( speakable.speak() );
speakable = puppy;
System.out.println( speakable.wag() );
System.out.println( speakable.speak() );
System.out.println( puppy.wag() );
System.out.println( puppy.speak() );
System.out.println( kitty.sleep( 5 ) );
System.out.println( kitty.speak() );
```

The keyword to inherit from an abstract class is _____________________

The keyword to inherit from an interface is _____________________
In the statement

```java
new FilledOval( 30, 30, 100, 100, canvas );
```

the first 4 arguments represent ______

(write the letter representing your answer in the blank above.)

What is the output of this recursive method if it is invoked as `ref.mystery( 11 );`? Draw Stack Frames to help you answer this question.

```java
int mystery( int a ) {
    int b = a - 3;
    if ( b >= 5 ) {
        System.out.println( a + " " + b );
        a = b - mystery( b + 1 );
    } else {
        System.out.println( "Stop" );
        b = a + 2;
    }
    System.out.println( a + " " + b );
    return a + b;
}
```

In the current programming assignment, the constructors are to perform deep vs. shallow copies. Fill in the blanks to implement the public `Circle(Point center, int radius) { … }` constructor with deep copy. Remember to use the appropriate accessor/mutator methods.

```java
public Circle( Point center, int radius ) {
    // Invoke superclass ctor with name of this shape
    super( center, radius );
    this.setRadius( radius );             // Initialize radius member in this new Circle
    // Initialize center point in this new Circle
    ___________________________________________________________________
}
```

Fill in the blanks for the recursive `reverse()` method you implemented in the previous programming assignment.

```java
public void reverse( int[] array, int low, int high ) {
    if ( ____________________ == null )
        return;
    if ( ____________________ >= 1 ) {    // perform swap and recursive call
        int tmp = array[low];
        array[low] = __________________________;
        __________________________ = tmp;
        reverse( array, ______________________, ______________________ );
    }
}
```

A) Center point of oval and x diameter and y diameter
B) Upper left corner and lower right corner of bounding box
C) Center point of oval and width and height of bounding box
D) Upper left corner and width and height of bounding box
E) Center point of oval and x radius and y radius