Signature _____ **CSE 11 Quiz 3 Fall 2010** **Name** _____

cs11f ____ **Student ID** _____

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

Given the following definition of class Thing1, what is the output of the Java application Question1?

```
public class Thing1
{
  private int count;

  public Thing1( int count )
  {
    this.count = count;
  }

  public int getCount()
  {
    return this.count;
  }

  public void setCount( int count )
  {
    this.count = count;
  }

  public String toString()
  {
    if ( this.count == 1 )
      return "one";
    else if ( this.count == 2 )
      return "two";
    else if ( this.count == 3 )
      return "three";
    else
      return "too many";
  }

  public static void swap1( Thing1 t1, Thing1 t2 )
  {
    Thing1 temp;

    temp = t1;
    t1 = t2;
    t2 = temp;
  }
}
```

```
public class Question1
{
  public static void main( String[] args )
  {
    Thing1 first = new Thing1( 4 );
    Thing1 second = new Thing1( 2 );

    System.out.println( first.toString() );
    System.out.println( second.toString() );

    Thing1.swap1( first, second );

    System.out.println( first.toString() );
    System.out.println( second.toString() );

    Thing1 third = new Thing1( 1 );
    Thing1 fourth = new Thing1( 3 );;
    second.setCount( third.getCount() );
    first = fourth;

    System.out.println( first.toString() );
    System.out.println( second.toString() );
    System.out.println( third.toString() );
    System.out.println( fourth.toString() );

    System.out.println(
               first.toString().equals( fourth.toString() )
                    );
    System.out.println(
               second.toString().equals( third.toString() )
                    );
    System.out.println( first == fourth );
    System.out.println( second == third );
  }
}
```

Output

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

Given the following definitions:

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

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

  public Puppy()
  {
    // ctor initialization here
  }

  public String speak()
  {
    return PUPPY_SPEAK;
  }

  public void sleep( int time )
  {
    // puppy sleeps for time seconds
  }
}
```

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

  public Kitty()
  {
    // ctor initialization here
  }

  public String speak()
  {
    return KITTY_SPEAK;
  }

  public void wag()
  {
    // kitty wags its tail
  }
}
```

And the following variable definitions:

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

Indicate which are valid Java statements. Consider each statement executed sequentially in the order it appears.

A) Valid Java statement – No Compiler Error
B) Invalid Java statement – Compiler Error

Hint: What does the compiler know about any reference variable at compile time (vs. run time)?

```
puppy = new Puppy();          _______

puppy.speak();                _______

puppy.wag();                  _______

puppy.sleep( 1000 );          _______

kitty = new Kitty();          _______

kitty.speak();                _______

kitty.wag();                  _______

kitty.sleep( 2000 );          _______

speakable = puppy;            _______

speakable.speak();            _______

speakable.sleep( 3000 );      _______

speakable = new Speakable();  _______

speakable = kitty;            _______

speakable.speak();            _______

speakable.wag();              _______

puppy = kitty;                _______
```