Signature _____ **CSE 11 Quiz 3 Fall 2008** **Name** _____

cs11f ____ **Student ID** _____

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

When designing an Object-Oriented Program with objects somewhat modeling the real world, we often start by listing properties and behaviors and deciding what is public and what is private. Use these underlined words to fill in the first 5 blanks below:

Object _____ will be modeled as instance variables, constants, static variables. By default,

we usually make these _____ so no other code outside of the class in which they are defined has direct access to them.

Object _____ will be modeled as constructors and methods. Usually these are

_____ so they are part of the object's interface that other objects can use versus

_____ which means they are for internal use only.

Which GUI component covered in Ch. 11 of the textbook is the most useful for getting a single line of input

from the user? _____

Which GUI component covered in Ch 11 of the textbook is the most useful for displaying multiple lines of text?

_____

The name of the event handler method used to handle ActionEvents is _____.

What is returned by each of the following method invocations when `bigString` is

```
"I drank java on the island of Java."
```

```
bigString.charAt( 3 )                          _____
bigString.charAt( bigString.length() - 1 )     _____
bigString.substring( 5, 11 )                   _____
bigString.indexOf( "java" )                    _____
```

(over)

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 wag()
  {
    // wag the tail
  }
}
```

```
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 sleep( int time )
  {
    // kitty sleeps for time seconds
  }
}
```

And the following variable definitions:

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

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

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

```
puppy = new Puppy();          _______

kitty = new Kitty();          _______

puppy.speak();                _______

puppy.wag();                  _______

puppy.sleep( 1000 );          _______

kitty.speak();                _______

kitty.wag();                  _______

kitty.sleep( 2000 );          _______

speakable = puppy;            _______

speakable.speak();            _______

speakable.wag();              _______

speakable = kitty;            _______

speakable.speak();            _______

speakable.sleep( 3000 );      _______

puppy = kitty;                _______

speakable = new Speakable();  _______
```

```
int result = 20;
int count = 0;

while ( count < 15 )
{
  count++;
  --result;
}
```

Value of count after loop terminates

_____

Value of result after loop terminates

_____