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:

```java
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:

```java
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

```java
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();
```

```java
int result = 20;
int count = 0;
while ( count < 15 )
{
    count++;
    --result;
}
```

Value of count after loop terminates

Value of result after loop terminates