James J. Kim
Homework 2

3.2 Suppose the class *Sub* extends *Sandwich*. Which of the following statements are legal?

Sandwich x = new Sandwich();
Sub y = new Sub();

(a) x = y; *Legal*
(b) y = x; Not *Legal*
(c) Sub y = new Sandwich(); *Not Legal*
(d) Sandwich x = new Sub(); *Legal*

3.6 Explain what an overloaded method is and give an example.

A good example of overloaded method is a class constructor. Class constructor can have multiple constructor methods with different method signatures.

Example:
public Box() {}:
public Box(int length) {};

3.7 Explain what an overridden method is and give an example.

An overridden method is used when inherited method from the parent class doesn't meet the needs of a subclass. You are using the same method name PLUS the same method signature but changing the implementation.

Example:
public void sayHello() { System.out.println ("Saying hello !");
public void sayHello() { System.out.println ("Saying hello to all !");

3.8 Explain what accidental overloading is and the preferred Java method for preventing it.

This occurs when you attempt to override the inherited method from the parent/ super class but the method signature is different. You can prevent this by utilizing @Override attribute when implementing a method.

Example:
@Override
public void sayHello (String name) { … };

3.14 True or False? It is legal in a superclass for a method to overload a method in a subclass. Explain.

False - Super class doesn't have any knowledge of the methods in the subclass.

4.2 Write the Java code to declare a new class *Bee* which is a subclass of *Insect*. The noise made by a *Bee* is "Buzz".

```
public class Bee extends Insect {
        @Override
        public void makeSound() {
                System.out.println("Buzz");
        }
}
```

4.3 Write the Java code to declare a new abstract class *Amphibian* that implements the *MakesNoise* interface.

```
public abstract class Amphibian implements MakesNoise {
                        …
}
```

4.4 Write the Java code to declare a new class *Frog* which is a subclass of *Amphibian*. The noise made by a *Frog* is "Ribbet".

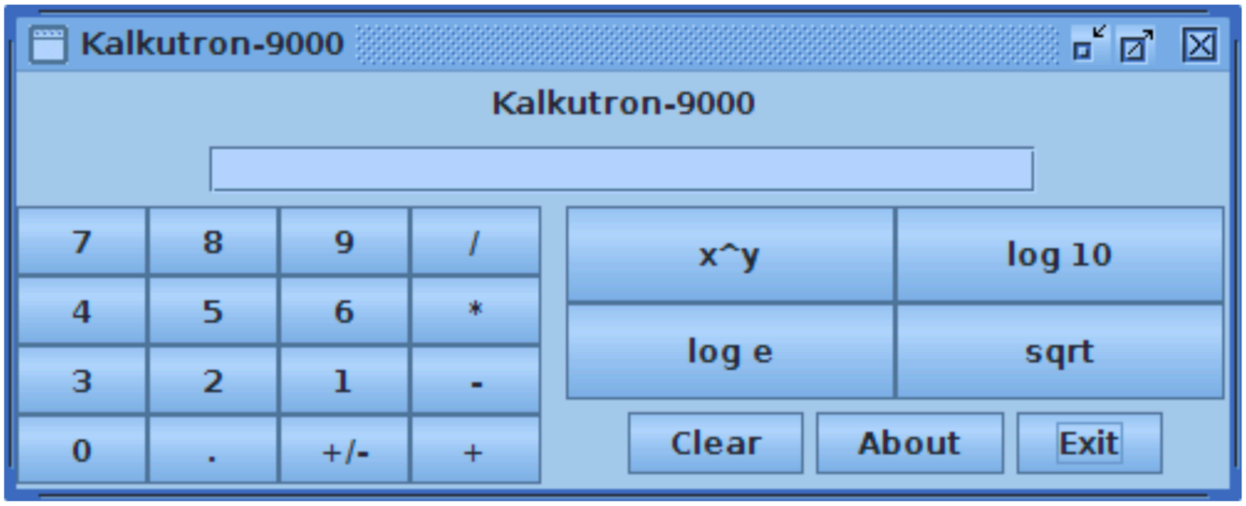
```
public class Frog extends Amphibian {
        @Override
        public void makeSound() {
                System.out.println("Ribbet");
        }
}
```

4.5 Modify the *run*() method of *Main* and add some *Bee*s and *Frog*s to *critters*. Build your program and verify that it works correctly. Include all of your .java source code files in the zip archive that you submit for grading.

Source Files Attached

5.1 For these exercises, include your completed .java files in the zip archive that you submit for grading. Complete the code in the provided *View* class to implement this GUI interface for a calculator. The calculator does not have to be fully functional; the primary objective of the assignment is to implement the GUI.

Source Files Attached

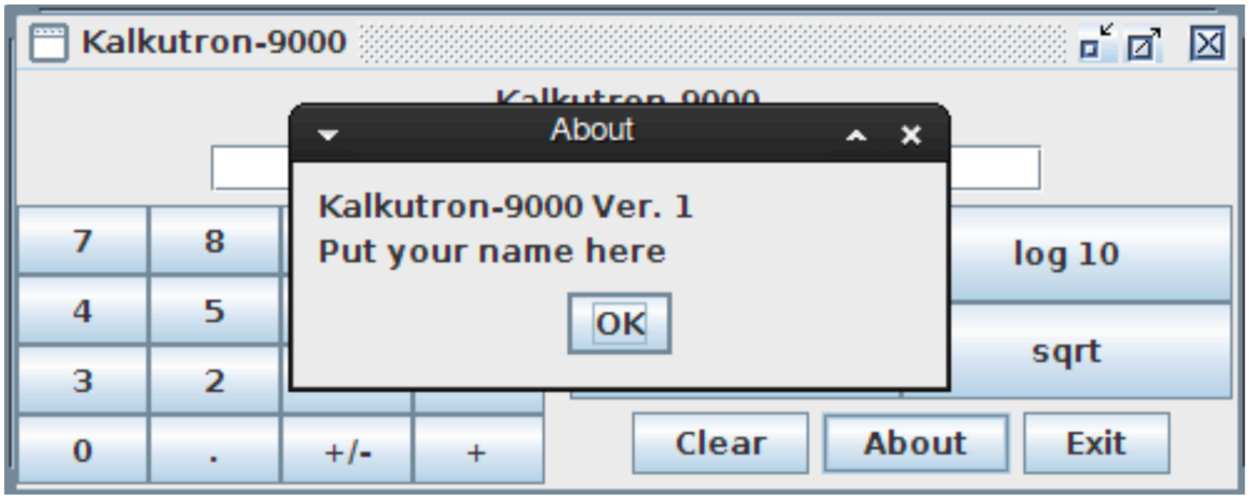


5.2 Complete the code in *actionPerformed*() so when the Exit button is clicked, the application will terminate.

Source Files Attached

5.3 Complete the code in *actionPerformed*() so when the About button is clicked, the application will display this about dialog:

Source Files Attached



Kalkutron-9000 Ver. 1
Put your name here

6.2 Explain how a local class differs from an inner class.

Local classes are declared within a block where it’s often times the only place where its needed and inner class is declared inside of another class.

6.3 Explain how an anonymous class differs from an inner and local class.

Anonymous class doesn’t have a name and it’s not considered to be a class. Its considered more to be an expression that can be passed around, even as a method parameter.