1 Submission Instructions

Create a document using your favorite word processor and type your exercise solutions. At the top of the document be sure to include your name and the homework assignment number, e.g. HW2. Convert this document into **Adobe PDF format** and name the PDF file *<asuriteid>.pdf* where *<asuriteid>* is your ASURITE user id (for example, my ASURITE user id is *kburger2* so my file would be named *kburger2.pdf*). To convert your document into PDF format, Microsoft Office versions 2008 and newer will export the document into PDF format by selecting the proper menu item from the File menu. The same is true of Open Office and Libre Office. Otherwise, you may use a freeware PDF converter program, e.g., *CutePDF* is one such program.

Next, create a folder named *<asuriteid>* and copy *<asuriteid>.pdf* to that folder. Copy any Java source code files to this folder (note: Java source code files are the files with a *.java* file name extension; **do not** copy the *.class* files as we do not need those).

Next, compress the *<asuriteid>* folder creating a **zip archive** file named *<asuriteid>.zip*. Upload *<asuriteid>.zip* to the Homework Assignment 2 link by the assignment deadline. The deadline is in course schedule info. Consult the online syllabus for the late and academic integrity policies.

Note: not all of these exercises will be graded, i.e., random ones will be selected for grading.

2 Learning Objectives

1. Properly use the public, private, and protected accessibility modifiers.
2. Write Java code to override and overload methods.
3. Recognize when inheritance is present among classes in an OOD.
4. Implement classes using inheritance.
5. To recognize when polymorphism is present in an inheritance hierarchy and to implement it.
6. To declare and implement a Java interface.
7. To implement a GUI.

3 Objects, Classes, and Inheritance

3.1 Is it required to provide an accessor and/or mutator method for every instance variable of a class? If yes, explain why this is required, and if no, explain why not.

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;
(b) y = x;
(c) Sub y = new Sandwich();
(d) Sandwich x = new Sub();
```

3.3 True or False? A subclass declaration will generally contain declarations for instance variables that are specific to object of that subclass, i.e., those instance variables represent attributes that are not part of superclass objects.

3.4 True or False? A superclass declaration will generally contain declarations for instance variables that are specific to objects of that superclass, i.e., those instance variables represent attributes that are not part of subclass objects.

3.5 Consider classes *C1*, *C2*, and *C3*. Answer the following questions.

```
class C1 {
   public int x1;
   protected int x2;
   private int x3;
   public void c1Method1() {}
   protected void c1Method2() {}
   private void c1Method3() {}
}
```

```
class C2 extends C1 {
    public int y1;
    protected int y2;
    private int y3;
    public void c2Method1() {}
    protected void c2Method2() {}
    private void c2Method3() {}
}
```

```
class C3 extends C2 {
    public int z1;
    protected int z2;
    private int z3;
    public void c3Method1() {}
    protected void c3Method2() {}
    private void c3Method3() {}
}
```

1. Which instance variables x1, x2, x3 declared in C1 are directly accessible in c1Method1()?
2. Which instance variables x1, x2, x3 declared in C1 are directly accessible in c1Method2()?
3. Which instance variables x1, x2, x3 declared in C1 are directly accessible in c1Method3()?
4. Which instance variables x1, x2, x3 declared in C1 are directly accessible in c2Method1()?
5. Which instance variables x1, x2, x3 declared in C1 are directly accessible in c2Method2()?
6. Which instance variables x1, x2, x3 declared in C1 are directly accessible in c2Method3()?
7. Which instance variables x1, x2, x3 declared in C1 are directly accessible in c3Method1()?
8. Which instance variables x1, x2, x3 declared in C1 are directly accessible in c3Method2()?
9. Which instance variables x1, x2, x3 declared in C1 are directly accessible in c3Method3()?
10. Which instance variables y1, y2, y3 declared in C2 are directly accessible in c1Method1()?
11. Which instance variables y1, y2, y3 declared in C2 are directly accessible in c1Method2()?
12. Which instance variables y1, y2, y3 declared in C2 are directly accessible in c1Method3()?
13. Which instance variables z1, z2, z3 declared in C3 are directly accessible in c1Method1()?
14. Which instance variables z1, z2, z3 declared in C3 are directly accessible in c1Method2()?
15. Which instance variables z1, z2, z3 declared in C3 are directly accessible in c1Method3()?
16. Which instance methods c1Method1(), c1Method2(), c1Method3() are callable from c2Method1()?
17. Which instance methods c1Method1(), c1Method2(), c1Method3() are callable from c2Method2()?
18. Which instance methods c1Method1(), c1Method2(), c1Method3() are callable from c2Method3()?
19. Which instance methods c1Method1(), c1Method2(), c1Method3() are callable from c3Method1()?
20. Which instance methods c1Method1(), c1Method2(), c1Method3() are callable from c3Method2()?
21. Which instance methods c1Method1(), c1Method2(), c1Method3() are callable from c3Method3()?
22. Which instance methods c2Method1(), c2Method2(), c2Method3() are callable from c1Method1()?
23. Which instance methods c2Method1(), c2Method2(), c2Method3() are callable from c1Method2()?
24. Which instance methods c2Method1(), c2Method2(), c2Method3() are callable from c1Method3()?
25. How many instance variables are encapsulated within a C1 object?
26. How many instance variables are encapsulated within a C2 object?
27. How many instance variables are encapsulated within a C3 object?

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

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

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

3.9 If an overridden method in a subclass needs to call the overridden superclass method, how is this accomplished?

3.10 True or False? It is legal for a method in a class to overload another method also in the same class. Explain.

3.11 True or False? It is legal in a class for a method to override another method also in the same class. Explain.

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

3.13 True or False? It is legal in a subclass for a method to override a method in the superclass. Explain.

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

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

3.16 In a subclass constructor, the superclass default constructor is called automatically before the statements of the subclass constructor begin executing. Suppose we wish to call a different superclass constructor (i.e., not the default constructor) from the subclass constructor. Explain how this is accomplished and give an example.

3.17 Explain how an abstract class differs from a concrete class.

4 Objects, Classes, Polymorphism, and Interfaces

4.1 In the video lecture for *Interfaces : Section 6* we discussed an example program that implements an inheritance hierarchy (*Mammal* is the superclass of *Cat* and *Dog*; *Insect* is the superclass of *Cricket*). Which method or methods in that program are called polymorphically?

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".

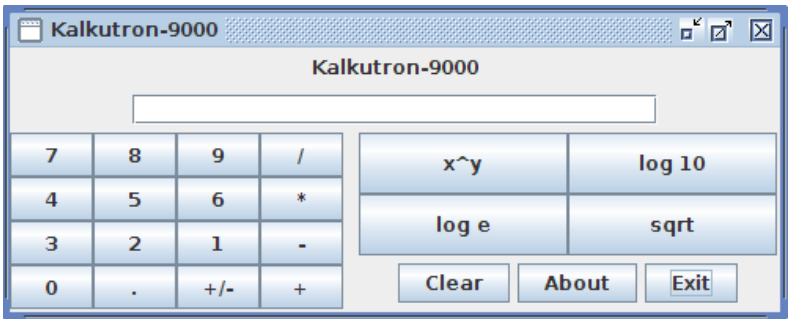
4.3 Write the Java code to declare a new abstract class *Amphibian* that implements the *MakesNoise* interface.

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".

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.

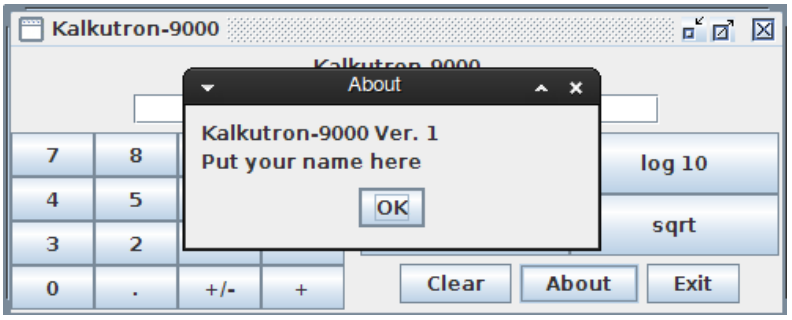
5 GUI Programming

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.



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

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



6 Nested Classes

6.1 Explain what an inner class is.

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

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