CSE231 Computer Programming (2)

Lab 05

1) What are the mistakes in the following program? Suggest a way to fix all errors and show the exact output after fixing it.

```
class A {
    public void f1(){
System.out.println("A.f1 called");}
public final void f2() {
System.out.println("A.f2 called");
}
 public abstract void f3();
}
class B extends A {
    public void f1() {
super.f1();System.out.println("B.f1 called"); }
  public void f2() {
super.f2();
System.out.println("B.f2 called"); }

 public void f3() {
System.out.println("B.f3 called");}
}
class C extends A {
    public void f1() { super.f1();
System.out.println("C.f1 called"); } }
abstract class D extends C {
    public void f3() {
System.out.println("D.f3 called");} }
final class E extends D {
}
class F extends E {
    public void f3() {
super.f3();
System.out.println("F.f3 called");} }
public class Test {
    public static void main(String[] args) {
A[] ps = new A[6];
ps[0] = new A();
ps[1] = new B();
ps[2] = new C();
ps[3] = new D();
ps[4] = new E();
 ps[5] = new F();

for (int i = 0; i < ps.length; i++) {
ps[i].f1();
ps[i].f2();
ps[i].f3();
        }
    }
}
```

expected output:

```
-------------------------
A.f1 called
b.f1 called
A.f2 called
b.f3 called
--------------------
A.f1 called
C.f1 called
A.f2 called
C.f3 called
--------------------
A.f1 called
C.f1 called
A.f2 called
D.f3 called
```

2) Assume we are interested in representing three Geometric shapes: (1) Line: with a start (javafx.geometry.Point2D) and end (javafx.geometry.Point2D). (2) Rectangle: with a start (javafx.geometry.Point2D), length (double) and width (double). (3) Ellipse: with a start (javafx.geometry.Point2D), length (double) and width (double).
   Define the appropriate classes. All classes should have a public constructor, accessors to all fields. All shapes should have a draw() method which is specific to every shape type, it should simply prints the shape information. For each class and each method you defined, specify whether it can be declared as abstract/final.

3) Create a Class that Represents a Canvas. A Canvas has multiple shapes (use ArrayList). The Class should have The following methods :
   - Add shape: Add new shape to the canvas.
   - Remove shape: remove a shape from the canvas.
   - Get shape: returns the closest shape to a given (javafx.geometry.Point2D) **hint:** you will have to calculate the distance between shape start point and the given point.
   - draw All : Simply draw all Canvas shapes.

   Write a simple driver program that test the functionality of the Canvas Class.

4) Define an interface called Moveable with a single method void move(double dx, double dy). Create a class that represents a new shape type called Diamond: with a start (javafx.geometry.Point2D), length (double) and width (double), your class should extends the Shape class defined in Problem2. Your class should also implements the Moveable and java.lang.Comparable interfaces. Write a simple driver program that test the functionality of the Diamond Class.

5) Create an array of Moveables and fill it with N diamond objects and with arbitrary values for the fields. Iterate through the array of Moveables and move each object in the array. Use the java.util.Arrays.sort() method on your array. And Finally iterate over your array and invoke the draw() method in every object.