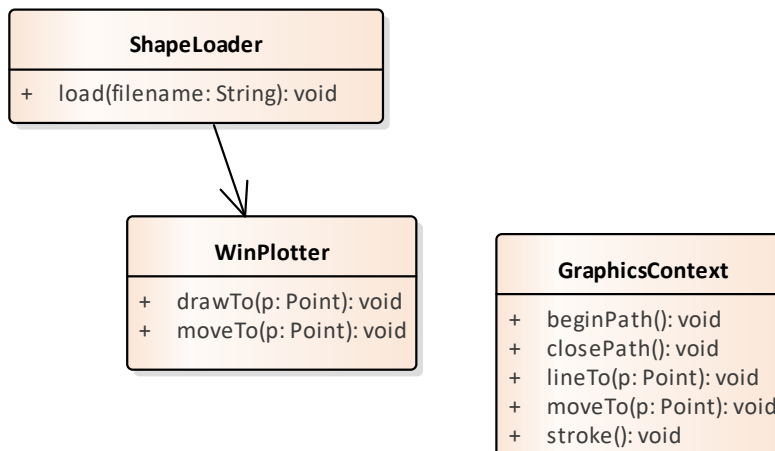# Half-Exam 1   Name:

No note-sheets, calculators, etc. on this exam. Please read the whole exam before you get started.

1. (30 points) **Edit** the UML diagram below to illustrate the transition from the old WinPlotter (based on Swing) to the new JavaFX GraphicsContext. **Edit** the UML diagram **in detail** to illustrate the Adapter Pattern.

ShapeLoader

+   load(filename: String): void

WinPlotter

+   drawTo(p: Point): void
+   moveTo(p: Point): void

GraphicsContext

+   beginPath(): void
+   closePath(): void
+   lineTo(p: Point): void
+   moveTo(p: Point): void
+   stroke(): void

2. (10 points) Consider a Time class with hours, minutes, and seconds fields. **Circle one**: an instance of this class is a **Domain Object** / an **Attribute. Explain** your choice.

3. (10 points) Consider a Refrigerator class with dimensions, weight, and efficiencyRating fields. **Circle one**: an instance of this class is a **Domain Object** / an **Attribute. Explain** your choice.

4. (10 points) ***Describe*** cohesion and coupling in your own words, making clear which is which.

5. (20 points) ***Give a code example*** illustrating low (poor) cohesion

6. (20 points) In the code and diagram below, the NullObject pattern is being applied to allow a Bicycle to have a NullKickstand or a Kickstand, but the pattern is not completely applied yet. ***Edit*** the code and diagram below to illustrate programming to an interface. Once your edit is complete, future kickstands should be able to be used without any changes to your solution.

```
public class Bicycle {

    private Kickstand kickstand;

    public void setKickstand(Kickstand k) {
        kickstand = k;
    }

    void mount() {/*…*/}
    void unMount() {/*…*/}
}
```

Bicycle

+ mount(): void
+ setKickstand(Kickstand): void
+ unMount(): void

Kickstand

+ kickDown(): void
+ kickUp(): void

«interface»
GenericKickstand

+ kickDown(): void
+ kickUp(): void

NullKickstand

+ kickDown(): void
+ kickUp(): void