CSSE 374: Lab 4-1 (Design Studio)

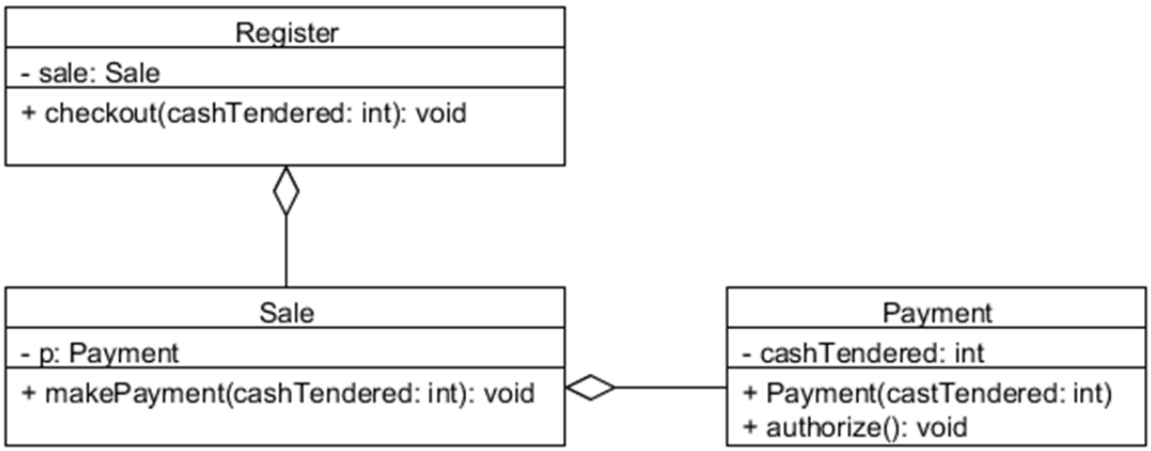
Download the **Lab4-1.zip** file from Moodle and import it as a Java project in your Eclipse IDE. Here are a couple of things you must be aware of throughout this exercise:

[Note 1] For some of the questions, you have to make assumptions about method names, signatures, and type of variables. It is alright to make those assumptions but make sure you document those assumptions both in your code snippets (using inline comments) and your design diagrams.

[Note 2] You **must not** have any syntax errors in your code snippets even though your code may not be functional.

Questions on Interaction Diagram (Reference Document)

1. Draw a UML class diagram based on the sequence diagram in the slide and append it to **docs/Answers.pdf**. [**5 points**]



2. Go to the **questions2** package and create code snippets that best reflect the SD in the slide incorporating your class diagram of Q1. [**5 points**]

```
public class Register {
	private Sale sale;

	public void checkout(int cashTendered) {
		// @TA:
		// 1. It is ok if students created a different method instead of checkout(int)
		// 2. It is also ok if the method did not have a parameter but student must pass
		// an int argument when calling makePayment
		sale.makePayment(cashTendered);
	}
}
```

```java
public class Sale {
	private Payment p;

	public void makePayment(int cashTendered) {
		p = new Payment(cashTendered);
		p.authorize();
	}
}


public class Payment {
	@SuppressWarnings("unused")
	private int cashTendered;

	public Payment(int cashTendered) {
		this.cashTendered = cashTendered;
	}

	public void authorize() {
	}
}
```

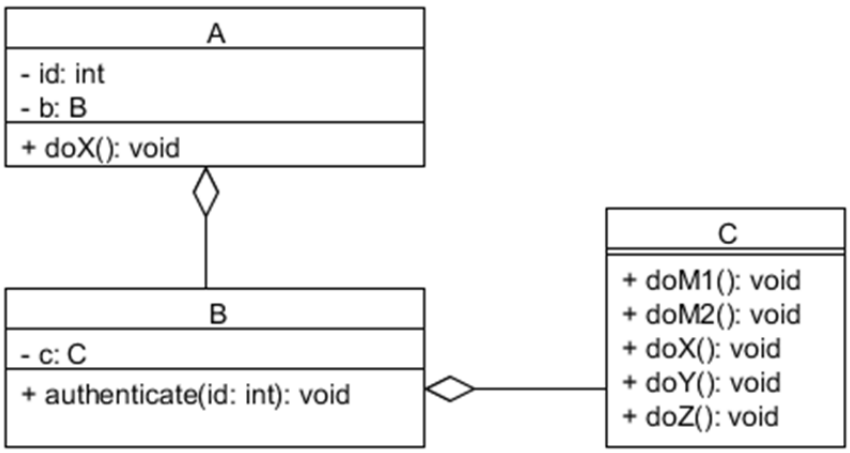
3. Go to the **question3** package in your Eclipse project and create necessary code snippets that best reflect the SD in the slide. [**5 points**]

```java
public class A {
	private int x;
	private B b;
	private C c;

	public void doX() {
		if(x < 10) {
			b.calculate();
		}
		else {
			c.calculate();
		}
	}
}

public class B {
	public void calculate() {

	}
}

public class C {
	public void calculate() {

	}
}
```

4. Go to the **question4** package in your Eclipse project and create necessary code snippets that best reflect the SD in the slide. [**5 points**]

```
public class Sale {
	private SalesLineItem[] lineItems;

	public int getTotal() {
		int t = 0;

		// @TA: while loop is also ok
		for(int i = 0; i < lineItems.length; ++i) {

			// @TA: This statement is absolutely necessary
			int st = lineItems[i].getSubtotal();

			// @TA: It is ok if this statement is missing
			t += st;
		}

		// @TA: They must have return t
		return t;
	}
}


public class SalesLineItem {
	public int getSubtotal() {
		// Returns sub-total
		return 0;
	}
}
```

5. Draw a UML class diagram based on the sequence diagram in the slide. Append it to **docs/Answers.pdf**. [**5 points**]



6. Go to the **question6** package in your Eclipse project and create necessary code snippets that best reflect the SD in the slide incorporating your class diagram of Q5. [**5 points**]

```
public class A {
	private B b;
	private int id;

	public void doX() {
		b.doA();
		b.authenticate(id);
	}
}

public class B {
	private C c;

	public void doA() {
		c.doB();
	}

	public void authenticate(int id) {
		c.doM1();
		c.doM2();

		c.doX();
		c.doY();
		c.doZ();
	}
}

public class C {
	public void doB() {}
	public void doM1() {}
	public void doM2() {}
	public void doX() {}
	public void doY() {}
	public void doZ() {}
}
```

7. Go to the **question7** package in your Eclipse project and create necessary code snippets that best reflect the SD in the slide. [**5 points**]

```
public class GameStarter {
	private Game g;

	public void startGame() {
		// @TA: They must instantiate either a Clock Thread or a Runnable Clock interface
		Clock c = new Clock();
		Thread runner = new Thread(c);
		runner.start();

		g.beginPlay();
	}
}

public class Game {
	public void beginPlay() {
	}
}
```

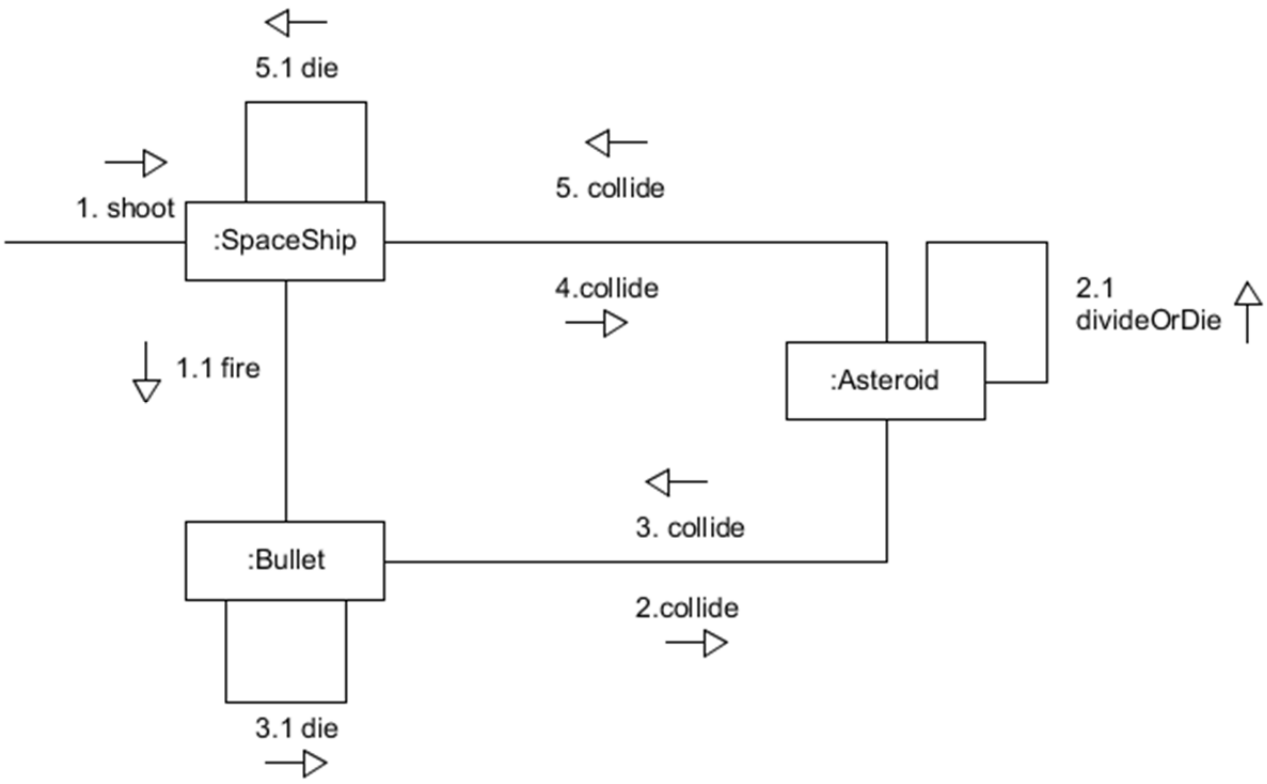
```
public class Clock implements Runnable {
	@Override
	public void run() {
	}
}
```

8. What is the key limitation of Communication Diagram? How do we overcome this limitation? Append it to **docs/Answers.pdf**. **[5 points]**

➔ It cannot represent the logic that has conditional checks within a loop. Hence, it is not good for automated code generation. It should be used only for **sketching** the overall interaction between objects. We overcome this limitation by using Sequence Diagram.

Design Exercises

9. Consider the following Asteroid arcade game: http://www.learn4good.com/games/action/asteroids_online.htm. Using a communication diagram, illustrate how a space ship, an asteroid, and a bullet (or missile) would interact with each other. Append the diagram to **docs/Answers.pdf**. [**10 points**]



Deliverables

Bundle your project in the **zip** format and upload it to Moodle.