

# Assignment 4

Reflection &amp; JET

**Assignment Due: Friday 20<sup>th</sup> 2017 @11:59pm\***

This assignment is worth 5% of your overall mark.

Please hand in a **single ZIP file** containing all deliverables to the assignment dropbox before this time. Failure to follow these instructions may result in your assignment not being marked!

---

## Getting Started

On [Canvas](#) where you downloaded this assignment handout, you'll also find a Zip archive called `se254-part2-assignment4.zip`. Firstly, make sure that **EMF** is installed for your version of Eclipse. You may find details on this here: <https://eclipse.org/modeling/emf/>. Once that's done, import the provided project into Eclipse by using the following:

- File → Import → Existing Projects into Workspace
- Choose "Select archive file" and browse to the downloaded Zip archive.

You'll see an Eclipse project called **SE254Assignment4** with three packages (one for each assignment question) and a **templates** folder where the JET templates you'll need to complete for the assignment will go. There are several classes provided – most are Main classes which are to be run to test each question, and example classes (such as Counter) which can be used to test your implementation. **se254.a4.q2.ImplementorTemplate** and **se254.a4.q3.ExtractorTemplate** are the classes generated by JET based on the templates **q2.javajet** and **q3.javajet**, which you are to complete. They will automatically be updated as you edit the template files.

## Question One: Java Reflection (33%)

Write a console program that asks the user for a class name, loads that class and creates an instance of it. We assume that the class has a constructor without any parameters. Then, the program prints out the names and values of the public fields of the created object, and also a list of the public methods that do not have any parameters. The program should let the user choose a method and execute that method on the created object. Afterwards, the program should again show the public variables with their values and allow the user to choose a method, and so on. Use the following class to test your implementation:

```
public class Counter {  
    public int c;  
  
    public void increment() { c++; }  
  
    public void decrement() { c--; }  
  
    public void reset() { c = 0; }  
}
```

## Implementation Instructions

Users should be able to run the program by running the provided **se254.a4.q1.Q1Main** class.

*\* ... however, the ADB will remain open until Monday 23<sup>rd</sup> October and there will be no late penalty for assignments submitted during that time*

## Question Two: JET Generator (33%)

Write a JET template that receives an object of type `Class` as argument. The object should represent a Java interface. The template generates a class that implements the interface, i.e. provides methods for all the method signatures it defines. The name of the generated class should be `XImplementation` where `X` is the name of the argument interface. The methods in the generated class do nothing or only return constant values: 0 for `int` and `double`, `false` for `boolean`, and `null` for reference types. You do not need to consider any other return types.

### Example

Consider the following interface `A`:

```
public interface A {  
    void m1(int x, int y);  
  
    int m2(Object a);  
  
    Object m3();  
}
```

The following class should be generated:

```
public class AImplementation implements A {  
  
    @Override  
    public void m1(int p1, int p2) { }  
  
    @Override  
    public int m2(Object p1) { return 0; }  
  
    @Override  
    public Object m3() { return null; }  
}
```

### Implementation Instructions

The template should be created in `q2.javajet`, in the `templates` folder of the provided Eclipse project. You can test your JET template by running the provided `se254.a4.q2.Q2Main` class.

### Hint

The unqualified (simple) name of a type can be acquired by using the `getSimpleName()` method on the corresponding `Class` object.

## Question Three: JET Generator (34%)

Write a JET template that receives an argument of the following type:

```
public class ExtractorArgument {  
    public String className;  
    public Class<?> a;  
    public Class<?> b;  
}
```

The template generates a new class with name `className`, which contains the public fields that classes `a` and `b` have in common. This means that the generated class has the public fields that are defined in `a` and `b` with the same type and name. You may neglect field modifiers such as `static`, i.e. they need not be compared or generated.

### Example

Consider the following classes `A` and `B`:

```
public class A {  
    public int x;  
    public String y;  
    public double z;  
}
```

```
public class B {  
    public int a;  
    public String y;  
    public int z;  
}
```

The following class `C` will be generated:

```
public class C {  
    public String y;  
}
```

### Implementation Instructions

The template should be created in `q3.javaJet`, in the `templates` folder of the provided Eclipse project. You can test your JET template by running the provided `se254.a4.q2.Q3Main` class.

### Submission Instructions

Please submit a single Zip file to the assignment dropbox **on or before 11.59pm, Friday October 20<sup>th</sup>, 2017** \*. The Zip file should be named `<yourupi>.zip`, and should contain the complete Eclipse project as you were given (see “Getting Started”), with all your modifications.

**Failure to follow these submission instructions may result in your assignment not being marked.**