Revision Questions (Lessons 1 - 3)

The code below has error(s). Suggest how to rectify it.

```
class A{

    class B{
        static void methodB(){
            System.out.println("Method B");
        }
    }
}
```

seems right, can use A.B.methodB()

How would you access the variable quack?

```
class Pond{

    static class Duck{
        int quack = 111;
    }
}
```

new Pond.Duck().quack

What is wrong with this code?

```
class A{

    String s = "AAA";

    void methodA(){
        System.out.println(s);
    }

    static class B{

        void methodB(){
            methodA(); methodA() MUST be static
        }
    }
}
```

static inner class cannot access enclosing class instance variables/methods

What are the statements that will be printed out?

```
public class TryCatchMystery {
   public static void main (String[] args) {
       try {
           method1();
           method2();
       } catch (IllegalArgumentException e) {
           System.out.println("main IllegalArgumentException");
       } catch (RuntimeException e) {
           System.out.println("main RuntimeException");
       }
   }
   public static void method1() {
       System.out.println("entered method1");
       try {
           method2();
       } catch (IllegalArgumentException e) {
           System.out.println("method1 IllegalArgumentException");
       }
       System.out.println("exited method1");
   }
   public static void method2() {
       System.out.println("entered method2");
       throw new IllegalArgumentException();
   }
}
```

"entered method1"
"entered method2"
"method1 IllegalArgumentException"
"exited method1"

"entered method2"
"main IllegalArgumentException"

Study the following code.

```
class Pond{

   private double size;
   private String type;

   private Pond(double size, String type){
       this.size = size;
       this.type = type;
   }
}
```

Without changing the code above (i.e. you may add code but not modify existing code), suggest **two** ways in which you could still allow Pond to be instantiated.

```
public static Pond makePond(double size, String type){
   return new Pond(size, type);
}


static class PondBuilder{
   private double size;
   private String type;

   PondBuilder() {
   }

   public PondBuilder setSize(double size){
      this.size = size;
      return this;
   }

   public PondBuilder setType(String type){
      this.type = type;
      return this;
   }

   public Pond build(){
      return new Pond(size, type);
   }
}
```

Coding Practice (Static Inner Class, Static Factory Method)

Write a class **`QuadraticEquation`** whose objective is to generate a quadratic equation with positive integer coefficients that have two real roots. The quadratic equation is represented by an object of the class **`QuadraticCoefficient`** that is a static inner class of **`QuadraticEquation`**.

The class **`QuadraticEquation`** has two private integer fields, **`aMax`** and **`cMax`** that represent the largest value of the coefficients a and c.
These values are initialized by the constructor, which is private.
The only way users instantiate your class is by a static factory method called **`getEquationGenerator()`**.
This static factory method allows the user to specify the values of **`aMax`** and **`cMax`**.

There is one public instance method **`getTwoRoots()`** that returns an instance of the inner class **`QuadraticCoefficient.`**
This method

- Generates a random value of a between 1 and **`aMax`**, both values inclusive
- Generates a random value of c between 1 and **`cMax`**, both values inclusive
- Generates a random value of b less than 100 such that the quadratic equations will have two real roots.
- Returns an instance of **`QuadraticCoefficient`** with the values of a, b and c generated

It has a static inner class **`QuadraticCoefficient`** that has

- Three private integer fields **`a`**, **`b`** and **`c`**
- Getter methods for these fields
- toString is overridden to display the following:
  When a = 7, b = 24 and c = 19 toString() returns the string
  “`y = 7x^2 + 24x + 19`”
  `(the quotes are not displayed on the screen)`

```
public class QuadraticEquation {
    private Integer aMax;
    private Integer cMax;

    private QuadraticEquation(Integer aMax, Integer cMax) {
        this.aMax = aMax;
        this.cMax = cMax;
    }

    public static QuadraticEquation getEquationGenerator(Integer aMax, Integer cMax) {
        return new QuadraticEquation(aMax, cMax);
    }

    public QuadraticCoefficient getTwoRoots() {
        Random random = new Random();
        Integer aMax = random.nextInt(1, this.aMax + 1);
        Integer cMax = random.nextInt(1, this.cMax + 1);

        Integer b2 = (int) Math.ceil(Math.sqrt(4 * aMax * cMax + 1));

        Integer b = random.nextInt(b2, 100);

        return new QuadraticCoefficient(aMax, b, cMax);
    }

    static class QuadraticCoefficient {
        private Integer a;
        private Integer b;
        private Integer c;

        public Integer getA() {
            return a;
        }
        public Integer getB() {
            return b;
        }
        public Integer getC() {
            return c;
        }

        private QuadraticCoefficient(Integer a, Integer b, Integer c){
            this.a = a;
            this.b = b;
            this.c = c;
        }

        @Override
        public String toString() {
            return "y = " + a + "x^2 + " + b + "x + " + c;
        }
    }
}
```