2. Coding Standards on a Page (or three)

Opt - Indicates Optional Rule for Cambridge Courses.

```
/**
* MyJavaCodingStandards.java - © BAESystems Detica 2013
*
* Not Protectively Marked
*/

import java.util.List;
import java.util.ArrayList;

/**
 *Does cool things. Positively chilly in fact. My kids would say
 * the functionality is "epic". Probably with an American accent.
 * Thanks TV.
 */
class MyJavaCodingStandards {
    Logger log = LoggerFactory.getLogger(MyJavaCodingStandards.class);

    public static final int RANGE_MIN = 0;
    public static final int RANGE_MAX = 100;

    private AnotherClass classVariable;

    MyJavaCodingStandards(){
        classVariable = new AnotherClass();
    }

    public void randomNumber(Integer x){
        if( null == x ){
            throw new IllegalArgumentException("x cannot be null");
        }

        MyObject obj = classVariable.getDodgyObject();

        if( null == obj ){
            throw new DodgyObjectException();
        }

        if( x > RANGE_MIN && x < RANGE_MAX ){
            ...
        }
    }


    public void avoidAutoboxing(int x){
        Integer intToAutoBox = x;

        int i = intToAutoBox;

    public static float methodExample(int x){
        int [] a = new int[20];

        float floatValue = (float) x;

        return floatValue;
    }
```

All code files must have a header with copy-right and protective markings Opt

Find More Detail in Appendix C.1.5

Imported classes should listed explicitly, avoid java.util.*

C.7.3

All code files must have a description Opt

C.1.5

CamelCase Class names

Constant—All caps with "_" separators. Avoid magic numbers, declare named constants.

B.5.3

C.2.7

Member variables should be declared private. Use getter and setter methods.

C.11.4

C.5.3

Avoid TAB, Use 4 Spaces Opt

All input parameters shall be checked for null values. Opt
Similarly, if you call a method that returns an object.

B.1.1

Code Defensively!

All input parameters shall be checked for range validity Opt

B.1.3

Be wary of automatic features such as autoboxing. Null is probably equivalent to 0 in most cases. But the designers of java cannot know that for sure so it is set to null. Autoboxing will throw an NullPointerException if 'x' is null as it will call x.intValue();

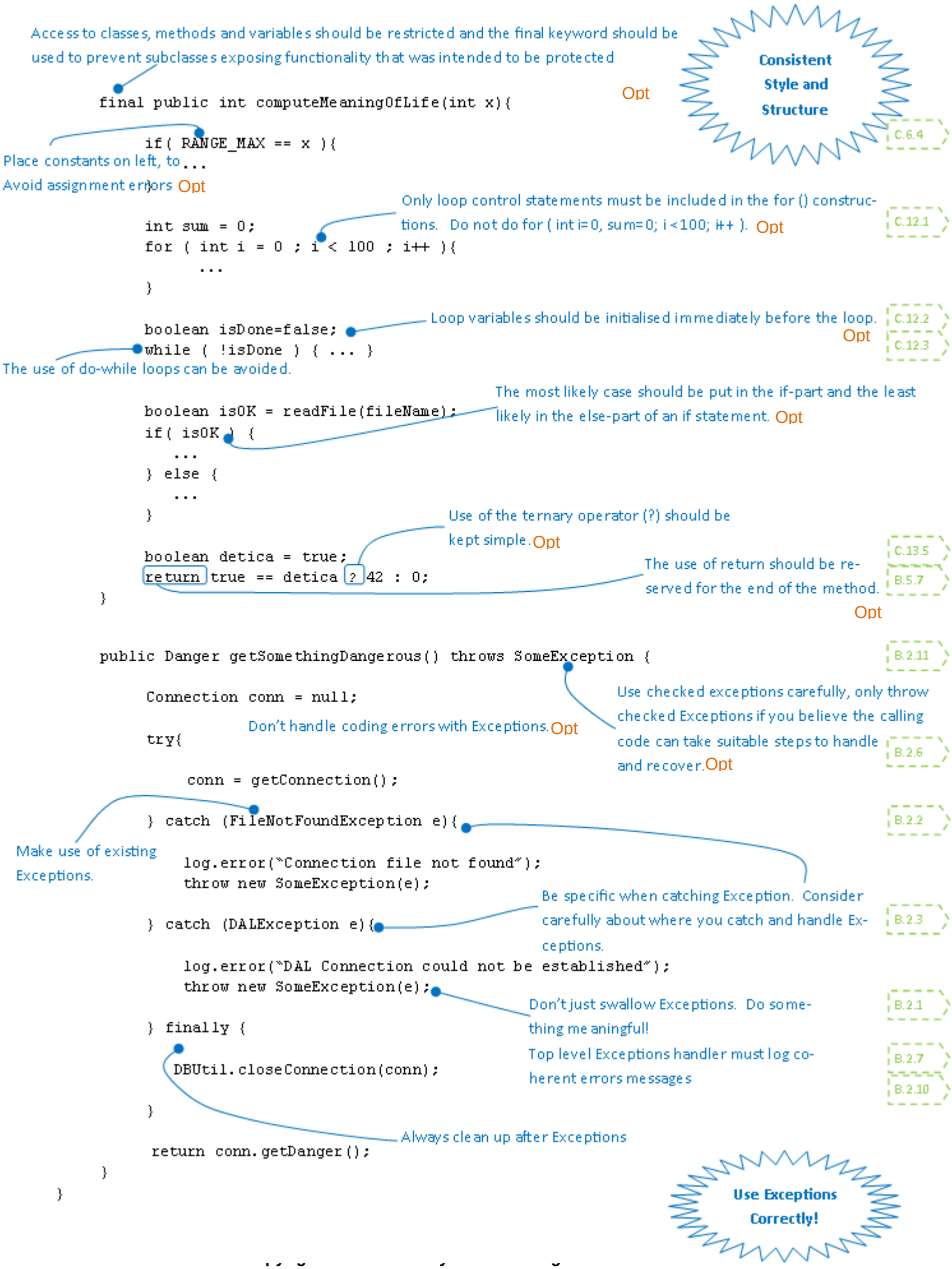
B.1.2

Use this method modifier order, avoid 'static public float'

Array specifiers must be attached to the type not the variable. Opt

C.10.2

Type conversions must always be done explicitly. Never rely on implicit type conversion.



```
import java.util.List;
import java.util.ArrayList;

public class DatabaseStuff extends Database implements Cloneable, Serializable {

    public executeQuery(String dodgyUserInput){
        ...
        Statement st = null;
        rs = st.exectueQuery("Select * from Table where x =" + dodgyUserInput);
        String result = fetchOneRowOnly(rs);
        ...
    }

    /**
     * Return lateral location of the specified position.
     * If the position is unset, NAN is returned
     *
     * @param x    X coordinate
     * @param y    Y coordinate
     * @param zone Zone of position
     * @return Lateral location
     * @throws IllegalArguementException If zone is <= 0
     */
    private double computeLocation(double x, double y, int zone)
                                    throws IllegalArguementException{
        switch (condition){
            case 1 :
                statements;
                /* Falls through */
            case 2 :
                statements;
                break;
            default :
                statements;
                break;
        }

        Matrix4x4 matrix;
        double    cosAngle;
        double    sinAngle;

        // Create a new identity matrix
        matrix = new Matrix4x4();

        // Precompute angles for efficiency
        cosAngle = Math.cos(angle);
        sinAngle = Math.sin(angle);

        // Specify matrix as a rotation transformation
        matrix.setElement(1, 1, cosAngle);
        return 0.0;
    }
}
```

The class and interface declarations should have the following form.

C.2.15

Use prepared statements. Allowing the execution of any String passed into the method leave the Database completely vulnerable to exploit.

Opt

Think about Security

B.4

Think about Performance

Think about Performance, think how the code will perform on the real datasets. For Example, don't select all then filter, use correct SQL.

Opt

C.4.3

JavaDoc comments should have the following form.

Opt

B.2.9

C.5.7

The switch statement should have the following form.

A comment should be added when no break is include.

Every switch statement should include a default case.

Opt

Variables in declarations can be left aligned.
Variable names must be mixed case starting with lowercase.

C.6.4

C.6.2

Logical units within a block should be separated by one blank line and commented correctly.

Opt

C.4.5

C.4.6

Use // for all non-JavaDoc comments, including Multi-line comments. The comments should be indented relative to their position in the code.

Opt

Remove all commented out code

C.1.1

C.1.4

Files should not exceed 1000 lines in length. Do not include more than one class per file. File content must be kept within 120 columns.