

## **ASSIGNMENT SUBMISSION STANDARD for CST8221 – Java Application Programming**

### **Note:**

When submitting your assignment, please replace the ***square-bracketed text*** with the required information. The percentage next to each submission item indicates that item's maximal mark. That value will be **deducted** if you do not comply with the standard.

### **A. Submission envelope label (up to 5%)**

Two types of assignment submission are required for this course: Assignment Box (AB) submission (printed paper) and Blackboard Digital (BBD) electronic submission. The contents of the AB and BBD submissions will be outlined in the assignment submission section of each individual assignment.

The Assignment Box submission must be placed into an unsealed page-size envelope labeled with the following information:

*[Student Name & ID number]*

**Course:** CST8221 – JAP, **Lab Section:** [301 or 302]

**Assignment:** *[number]*

**Professor:** *[place here the name of your lab professor]*

**Date:** *[submission date]*

**Note:** Students who submit their assignments in transparent envelopes need not label them. They must make sure that their cover page (see below) is visible.

### **B. Cover Page (up to 5%)**

All printed assignments must have a cover page with the following information:

*[Student Name & ID number]*

**Course:** CST8221 – JAP, **Lab Section:** [301 or 302]

**Assignment:** *[number]*

**Professor:** *[place here the name of your lab professor]*

**Due Date:** *[due date]*

**Date:** *[submission date]*

**Contents:** *[you should list here all the files/printouts submitted; they must be listed in the same order as the order they are arranged in the envelope]*

### **C. Test Plan (up to 10%)**

You must provide a thorough test plan describing your testing procedures and test cases. The test cases must include both normal and abnormal user input and run-time situations.

### **D. Coding Conventions (up to 10%)**

A submitted Java program (code and comments) should follow the conventions (except for the braces) described in the document “**Java Code Conventions**”. The document can be found at: <http://www.oracle.com/technetwork/java/javase/documentation/codeconvtoc-136057.html>

A submitted Java class program **must** contain file comments (file header), class doc comments, method doc comments, field doc comments, and implementation comments.

### D1. File Header (up to 5%)

The file comments must contain the following information:

**File name:** [ *YourMasterpiece.java* ]

**Author:** [ *Student name, ID#* ]

**Course:** CST8221 – JAP, **Lab Section:** [301 or 302]

**Assignment:** [ *number* ]

**Date:** [ *the date of the final version of the file* ]

**Professor:** [ *place here the name of your lab professor* ]

**Purpose:** [ *brief description of the contents* ]

**Class list:** [ *include this only if there is more than one class in the file* ]

### D2. Class Doc Comments (up to 5%)

Each class definitions must have a **Java doc comment** containing the following information:

[Brief description of the purpose of the class]

**@author** [ *your name if you wrote the class* ]

**@version** [ *version number* ]

**@see** [ *package name or a class name – for example, java.io, java.lang.String* ]

**@since** [ *the version number (1.8\_XX) of the Java compiler used to compile the class* ]

### D3. Method Doc Comments (up to 5%)

Each of the class method definitions must have a **Java doc comment** containing at least the following information:

[Brief description of the purpose of the method]

**@param** [ *name description* ] – one per parameter and only if there are parameters

**@return** [ *type description* ] – used only if there is a return type different from void

### D4. Field Doc Comments (up to 5%)

Each of the class fields (data members) must have a **Java doc comment**, which describes briefly the purpose of the field (variable). If the field is final, the **{@value}** tag must be used.

### D5. Implementation Conventions and Comments (up to 5%)

- All local variables must be commented. The comment must explain the use of the variable.
- All function segments must be commented (a segment is a sequence of related statements i.e. loops, switches, if-else ladders and sequence of linear statements performing some distinctive task.)
- Each important line of code must be commented. A student should use their judgment to decide whether the line is important. Important lines: testing some special conditions; complex calculations; conversions and so on. Do not comment overly your programs.
- The coding style and the naming should follow the Java coding conventions.

All assignment submissions must be in printed form. No assignment can include hand-written material or corrections unless explicitly exempted by the professor in the written assignment description.

If the submission, paper or digital, cannot be read, it will not be marked.

# **ASSIGNMENT MARKING GUIDE**

## **for CST8221 – Java Application Programming**

### **Definition of Working Program**

A Java program is a working program if and only if

- All classes in the program compile with no errors with 1.8\_45 or later **javac** compiler;
- The program runs and does not crash at run-time;
- The program complies with the functional specifications given in the assignment;
- The program produces the expected output as specified in the assignment. If output or output files are provided for the assignment, the program must produce identical output files when it is run with the corresponding test files.

The following set of marking rules will be applied when your assignment marks are determined.

### **RYGI. “You are Great and Innocent” Rule**

At the beginning you receive 100% of the maximal mark for the assignment.

And then some deductions might apply (We hope that will never happen).

### **RBGZ. “You Start from Big Gargantuan Zero” Rule**

If you do not submit a compiling or working program, you automatically turn into a winner of the “Big Gargantuan **0**” award. You might still receive some credits but your final assignment mark will not exceed 50% of the maximal mark for the assignment.

### **RWI. “What IF” Rule**

Your program will be tested thoroughly in attempt to find all program “deficiencies”. If your program passes all the tests, the **RYGI** will be applied.

If some problems are found, depending on the problem some deductions will apply but no more than 40% of the maximal assignment mark.

### **RTIAOMB. “There is Always One More Bug” Rule**

Your source code may be carefully reviewed and if minor mistakes are found, some deductions may apply but no more than 10% of the maximal mark. Minor mistakes are: redundant code, unused code, unnecessary extra steps, wrong comments, and alike.

### **RIDTS. “I Detest the Standards” Rule**

This rule will be applied if your assignment does not comply with the Assignment Submission Standard outlined above. The deductions are listed next to the corresponding requirements.

### **RITN. “Isn’t that nice” Rule**

You can get up to 10% for clever and clear coding, and useful enhancements.

## **RIAL. “I am Late” Rule (A Giant Zero again)**

Late assignments will receive a mark of zero. As stipulated in the course outline all programming assignments must be submitted in order to obtain course credit. Because of that you are strongly encouraged to submit on time even a non-working assignment so that you can obtain some partial marks.

## **RABGO. “Another Big Gargantuan O” Rule**

The code of the assignment must be originally written by the student submitting the assignment. If it happens that you have “borrowed” the entire code or even parts of the code, you will get 0. And something more. See the course outline. And this is serious.

In conclusion, here is the Java expression for calculating the assignment final mark:

```
BGZ = 0;  
finalMark = RYGI;  
if ( RABGO == false & RIAL == false)  
    finalMark - = (RBGO == false)? (RWI + RTIAOMB + RIDTS - RITN) : -someCredits  
else  
    finalMark = BGZ;
```

## **Important Note:**

Each programming assignment will be accompanied by a detailed marking sheet. The submission standard and the marking rules outlined above apply to your programming assignments only. The submission requirements and the marking rules for all other marked course work like lab exercises and hybrid activities will be clearly outlined in the submission section of the corresponding materials.

Enjoy and do not forget that:

***“Not everything that counts can be counted, and not everything that can be counted counts.”***

Albert Einstein