Assignment Guidelines

General Notes

- All assignments require you (the student) to develop source code using the ActionScript programming language.
- For each programming assignment, the criteria for developing source code and submitting your code are going to generally be the same. The only difference between assignments is the task to be implemented.
- Each programming assignment will be posted on moodle. It will clearly explain the tasks, submission instructions and due date.
- You will be given sufficient time to complete your assignment.
- **Assignments submitted after the due date will not be graded.**

Academic Honesty

- Assignments are NOT group projects. They must represent your own individual work.
- It is reasonable for students to consult or discuss the general solutions to an assignment. However, it is unreasonable for students to collaborate on detailed solutions, to copy code, or to give away code. If I can detect such occurrences from your code and comments, then the academic dishonesty will result in a course of action including a course grade of F and/or expulsion from the school.
- Note to students who give away their source code: Read up on that old saying that goes somewhat like this – "Don't provide fish, teach how to fish". If you don't have the time to explain the assignment task to your friend, there's myself and other tutors willing to spend as much time as required to help that person.
- Note to students who copy code: If you don't invest time and effort to solve the assignment, you are simply wasting your time and money.

Submitting Assignments

- In every assignment the number/name of files to be submitted will be specified.
- Assignments are to be submitted only through moodle. I will not accept any assignments submitted through email.
- Assignments may be submitted multiple times to moodle as long as you are still submitting before the due date (in other words, before the submission folder closes).
- Make sure to submit work only after checking (and re-checking, and re-checking) the specifications laid out in the assignment and of course following the style guide (File header, function headers, code fully commented, naming conventions, etc…).

- Moodle submissions steps:
  - ➢ Archive all files required into a single ZIP file (even if you are submitting only a single file).
  - ➢ The name of the ZIP file must follow this naming convention:

    ```
    <class_name>_<login>_<assignment number>.zip
    ```

    For example, if a student with login name foo.bar, submits assignment #3 for course CS176, the appropriate filename would be cs176_foo.bar_3.zip. Do not put the course section number in the filename.

    <u>Note:</u> The .zip file names are case-sensitive and must be in all lowercase, exactly as described above.

  - ➢ Let me repeat myself again: **Late assignments will not be graded.**

Grading

- It is your responsibility to test your program. In some assignments I will provide special test cases but you are also encouraged to create your own.

- Each assignment will be graded according to the breakdown described in the "Grading Breakdown" section below.

- For every early day the assignment is submitted the student will get 3 extra points.
  Example: If given 7 days to complete an assignment, submitting 5 days early with perfect implementation will lead to getting a grade of 115 on the assignment.

Grading Breakdown

Each assignment is worth a 100 points

SUB - Submission

1. Did not submit before the due date -100
2. Submitted a .rar/ .7z file *(Has to be .zip)* -100
3. No Proper zip file naming -100
4. No Proper .fla and .as files naming (according to the guideline) -100

NC - No Compile

1. Errors -100

OUT - Output (60)

1. Total is spread across all test cases.
   (If we have 3 test cases, each test case would be over 20)
2. Everything Wrong -60

Commenting (60)

1. FHC - File Header Comment (Max 20)
   (Every file submitted must have a file header with the correct format and information)
   - i. Incomplete or slightly incorrect format -5
   - ii. Seriously defective -10
   - iii. Missing entirely -20

2. FNC - Function Header Comment (Max 20)
   (Every function implemented by the student must have a header with the correct format and information)
   - i. Function with incomplete/incorrect header format -2
   - ii. A function without a header -5
   - iii. All functions without headers -20

3. ID - Internal Documentation (Max 20)
   - i. Uncommented imports -5
   - ii. Uncommented variable -5
   - iii. Severe lack of comments in function or none at all -5
   - iv. No internal commenting at all -20

CQ - Code quality (30)

1. Meaningful variable naming -5
2. Unused / Unneeded file imports -5
3. Must use curly braces -5
   (they are on a line by themselves)
4. All blocks must be indented properly -5
5. No white space between declarations and code -5
   (If done consistently, not just a single mistake)
6. Spaces after commas, and around binary operators -5

(If done consistently, not just a single mistake or typo)

- **<u>O - Other</u>**

    1. **This will be assignment specific**
    (To avoid getting points removed the student has to make sure to follow in details the instructions provided)