COMP 3761 Guidelines for Assignments

This document describes the general requirements and submission details for the course assignments in COMP 3761. These requirements are subject to change. Any clarifications or changes will be announced in class, and documented in a new version of this file. It is your responsibility to be aware of any assignment updates or changes.

1. You may choose to work with one partner or on your own. If you work in pair, each person is expected to contribute equally to each assignment.

2. For all assignment problems, you need to design your own algorithms and write code to implement the programs yourself. **Do not use code that you get from others or from the Internet**. If you submit code that you do not write, you shall face the disciplinary action as per BCIT policies.

3. The official programming language for this course is **Java**. You must use Java for all lab assignments.

4. You are free to use any of your favourite IDEs for your implementations. The instructor does not use any IDEs. Your submitted programs will be compiled using javac command with Java 7.

5. You are expected to write code that can be easily understood and adheres to sound principles of program design as taught in previous BCIT computing program courses. Since this course is about algorithms, in general, we will be evaluating the overall efficiency and sophistication of the algorithms and data structures that you use.

6. **For all algorithm design problems**, please do the following:

   - Design the algorithm and provide a description (preferably in pseudocode) of your algorithm.
   - Implement each algorithm in its own method. Use proper comments to explain your algorithm.
   - Write a single test driver (ie. one main method) for each assignment, unless specified otherwise.
   - Test your algorithm with legitimate inputs and show (and explain) your test results.
     Throughout this course, you may assume all test inputs are legitimate for the given problem.
   - Unless specified otherwise, the test program must read the inputs from standard input to allow input redirection from any ASCII text file. Do not pass in any input values as command-line arguments from the main test program. Do NOT hard-code any input values or hard-code any folder and/or file names for the test inputs in your programs.

Detailed requirements on Assignment Submission

1. Assignment is due at the due date and before the due time. Submissions received after the due date/time are marked as **late**.

   For each day (24hrs or portion thereof) the submission is late, 20% of the total marks for the assignment will be deducted. Submissions that are 5 or more days late will not be graded. There will be no exceptions to this policy, regardless of your reason.

2. If you work in pair, please submit a single copy of your deliverables with both names and student numbers. If multiple zip files are submitted for one assignment by the same individual(s), only the last submission (with the latest time stamp) will be marked.

3. For problems involving mathematical solving steps, you may handwrite (ensure your handwriting is legible) or word-process your solutions. All word documents submitted to SHAREIN should be in the **.pdf** or simple **.txt** (No .doc or .docx) file format.

4. For problems involving algorithm implementations, each source code file in your submission must include a header block listing the author(s) name(s), date, and description of the file contents. Other descriptive comments should be added as needed to ensure the program is easily understood by the others.

5. Print and submit a hard-copy of your solutions. Your test results should clearly show your sample test inputs and outputs for each program. Simply copy and paste the sample test results in a text file, and print out the text file. Please do **not** include the screen shots (or pictures) for the test outputs.

6. Upload a soft-copy of your solutions into the correct lab assignment folder under BCIT SHAREIN/COMP3761.

   Each assignment submission must be in one single zip file (in **.zip** format, **no .rar file**), which contains a folder named by your name(s). The naming convention for the folder name is your first and last name(s), in alphabetical order. For example, if Adam Smith was working with Tim Horton on one assignment, they would create a folder named as AdamSmith_TimHorton, which contains all the soft-copy of the solutions and the sample test files. This folder is then compressed as a single zip file and submitted as AdamSmith_TimHorton.zip.

   When the submitted zip file is unzipped, it should create a directory with your name(s). Within this directory, we expect to find all the source files required to build and run your program, including the sample test inputs and results, as well as any written answers or report (if required).

7. The program source files should only include the real java source code (.java files only). Please do **not** submit any of the following: the Java SDK files, IDE directories and files, the compiled Java class files or other compiler-generated files. Please do not use sub-folders in your submission. Do not use Java packages in your source code for the assignments in this course.

   Ensure all the source code in your submission folder can be compiled successfully by running the javac command ($ javac *.java).

Grading and Evaluation

1. Your program will be evaluated against the instructor's test input data. Some sample test data will be provided in advance, others may not.

   The test data is designed to make sure you understand and follow the problem requirements. Only legitimate input will be used. All tests will adhere to the input specifications for each problem. Make sure your program can handle the large (and legitimate) input sizes.

2. Solutions that include programs that do not compile or that compile but crash upon reading legitimate input data (either from standard keyboard or redirected from test file) will receive a grade of zero.

3. Marks will be deducted for programs that are poorly designed, and for algorithms that are confusing and difficult to understand.

4. In general, your solution will be graded according to this mark distribution:

   - Correctness of solution: 60%
   - Algorithm design and analysis: 30%
   - Ability to follow the general instructions: 10%