Assignment

The main purpose of this assignment is to introduce you to how assignments are handled in CO383. By the end of this assignment you will be familiar with the overall marking and evaluation system of CO383. And, you should be comfortable submitting assignments using the student UNIX environment.

For this assignment you are tasked with writing a command-line Java program that reads from the command line a single digit number and outputs its English name.

- Due date is as specified on the Moodle.
- Submission will be through the Raptor PC Environment.
- Go to your CO383 submission directory: /proj/co383c/assignment1/<login>
- Inside that directory create a file called submission1.java
- This is your assignment submission. This must be a Java program that:
  - Correctly compiles under the UNIX command line.
  - When run, reads a single digit non-negative number from the command line, and then outputs its English name in all small letters.
- Your program should write to standard out a single line with the correct output.The following example shows correct behaviour for your program (my userid is cd472):

```
cd472$ cd /proj/co383c/assignment1/cd472
cd472$ javac submission1.java
cd472$ echo 7 | java submission1
seven
```

- **Marking scheme:**
  - There are 5 total marks possible. You get 1 mark for successfully submitting your assignment on time.
  - Your program will be evaluated against a battery of 4 tests. You get 1 mark for each successful test, up to a total possible of 5/5.
  - You may assume all tested inputs are properly formed (i.e. every input is correctly formatted).
  - Your program will be evaluated automatically on the Raptor environment.

- **Further notes**
  - Read the marking scheme and submission instructions carefully, and make sure to follow them exactly.
  - Test your program!

- **Plagiarism and Duplication of Material**
- The work you submit must be your own. We will run checks on all submitted work in an effort to identify possible plagiarism, and take disciplinary action against anyone found to have committed plagiarism.
- Some guidelines on avoiding plagiarism:
  - One of the most common reasons for programming plagiarism is leaving work until the last minute. Avoid this by making sure that you know what you have to do (that is not necessarily the same as how to do it) as soon as an assessment is set. Then decide what you will need to do in order to complete the assignment. This will typically involve doing some background reading and programming practice. If in doubt about what is required, ask a member of the course team.
  - Another common reason is working too closely with one or more other students on the course. Do not program together with someone else, by which I mean do not work together at a single computer, or side by side, typing in more or less the same code. By all means discuss parts of an assignment, but do not thereby end up submitting the same code.
  - It is not acceptable to submit code that differs only in the comments and variable names, for instance. It is very easy for us to detect when this has been done and we will check for it.
  - Never let someone else have a copy of your code, no matter how desperate they are. Always advise someone in this position to seek help from their class supervisor or lecturer. Otherwise they will never properly learn for themselves.

    - It is not acceptable to post assignments on sites such as Freelancer and we treat such actions as evidence of attempted plagiarism, regardless of whether or not work is payed for.

- Further advice on plagiarism and collaboration is available from:
  http://www.cs.kent.ac.uk/teaching/student/assessment/plagiarism.local
- You are reminded of the rules about plagiarism that can be found in the Stage I Handbook. These rules apply to programming assignments. We reserve the right to apply checks to programs submitted for assignment in order to guard against plagiarism and to use programs submitted to test and refine our plagiarism detection methods both during the course and in the future.